SKU duplication on a unidirectional picking line

Desima Fivaz

Thesis presented in fulfillment of the requirements for the degree of
Master of Commerce (Operations Research)
in the Faculty of Economic and Management Science at Stellenbosch University

Supervisor: Prof SE Visagie

March 2013

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining and qualification.

owner of the copyright thereof (unless to the extent explicitly otherwise stated) and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

March, 2013

Abstract

PEP is a devision of Pepkor Retail Limited and is the biggest single brand store network in Southern Africa and also owns and runs the largest clothing factory in Southern Africa. It was founded in 1965 and has since grown to more than 1 400 stores in 9 African countries (there is a PEP store in almost every town and village in South Africa).

Currently the warehouse management system (WMS) implemented by PEP only allows for a stock keeping unit (SKU) to be placed on one picking line in one location when the distribution list (DBN) is released. Because pickers are only allowed to walk clockwise around the conveyor belt, they are forced to pass a location at least the same number of times as the number of branches to which the SKU is to be distributed to. Therefore if the SKUs with the highest pick frequency can be assigned to 2 locations (it is duplicating the SKU), the number of times each of these locations must be passed may be reduced.

In this study 4 questions are considered when 15 algorithms are constructed that will determine how an algorithm assign the SKUs to picking lines. Question 1 determines whether the original picking lines are to be treated separately (PS) or to combine them first (PC). The second question is to decide if the SKUs are first to be duplicated and then assigned to picking lines (DA) or if they are first assigned to picking lines and then duplicated (AD). Question 3 determines whether the non-duplicate and duplicate SKUs are treated separately (ND) or simultaneously (S) when they are assigned to the picking lines. The final question is to specify how the SKUs are assigned to the picking lines. Three assignment methods (cyclical, set length subset sequential assignment, remaining high, low cyclical assignment) and 6 clustering algorithms are introduced.

The conclusion is made that the SKUs with the highest pick frequency is duplicated first to yield the biggest saving in the number of cycles. Between 40–70% of the SKUs should be duplicated, dependant on the algorithm used. The only decision that has a major influence on the number of cycles is the assignment method used. Algorithm 5 and 8 yielded the greatest saving in the number of cycles (40.7% and 39.8% respectively), both implementing set length subset sequential assignment, followed by the clustering algorithms.

Uittreksel

PEP is 'n afdeling van Pepkor Retail Limited en is die grootste enkel-handelsmerk winkelnetwerk in Suidelike Afrika. PEP besit en bestuur ook die grootste klerefabriek in Suidelike Afrika. PEP is gestig in 1965 en het sedertien gegroei tot meer as 1 400 winkels in 9 Afrika lande (daar is 'n PEP winkel in amper elke dorp in Suid-Afrika).

Op die oomblik laat die pakhuisbestuurstelsel, wat deur PEP in sy distribusie sentrum geïmplementeer word, slegs toe dat voorraadeenhede (VEs) in 'n enkele vakkie langs 'n enkele uitsoeklyn geplaas word. Aangesien werkers slegs toegelaat word om kloksgewys om die vervoerband te beweeg, word hulle gedwing om ten minste soveel keer verby elke vakkie in die uitsoeklyn te loop as wat die aantal winkels is waarna die VEs in daardie vakkie versprei moet word. Dus indien die vakkies wat die VEs bevat wat na die meeste winkels versprei moet word, tussen 2 vakkies verdeel word (die VE word gedupliseer), verminder die aantal kere wat beide vakkies besoek moet word.

In hierdie studie word 4 vrae beskou wat geantwoord moet word wanneer die 15 algoritmes opgestel word, wat sal bepaal hoe die algoritme die VEs hanteer. Vraag 1 bepaal of die oorspronklike uitsoeklyne wat deur PEP verskaf is apart hanteer word en of hulle eers gekombineer moet word. Die tweede vraag bepaal of die VEs eers gedupliseer word en dan aan die onderskeie uitsoeklyne toegedeel word en of die VEs eers aan die uitsoeklyne toegedeel word en dan gedupliseer word. Vraag 3 is slegs van toepassing wanneer die VEs eers gedupliseer word en dan toegedeel word aan die uitsoeklyne, en bepaal of die nie-gedupliseerde en gedupliseerde VEs apart of gelyktydig hanteer word. Die laaste vraag spesifiseer met behulp van watter metode die VEs toegedeel word aan die onderskeie uitsoeklyne. Drie toedelingsmetodes (sikliese toedeling, vaste lengte subversameling opeenvolgende toedeling, oorblywende hoogste/laagste sikliese toedeling) en 6 bondelalgoritmes word voorgestel.

Die gevolgtrekking word gemaak dat die VEs met die hoogste uitsoek frekwensie eerste gedupliseer moet word om die grootste besparing mee te bring in die aantal siklusse om al die VEs uit te soek. Tussen 40–70% van die VEs moet gedupliseer word afhangende van die algoritme wat gebruik word. Die enigste besluit wat 'n noemenswaardige invloed op die aantal siklusse het is die toedelingsmetode. Algoritme 5 en 8 lewer die grootste besparing in die aantal siklusse (40.7% en 39.8% onderskeidelik), beide implementeer die vaste lengte subversameling opeenvolgende toedeling, gevolg deur die bondelalgoritmes.

Acknowledgements

Many people played a significant role in the work leading up to and during the writing of this dissertation. The author hereby wishes to express her deepest gratitude towards:

- Prof SE Visagie for his guidance, valuable insights, support and compassion during the compilation of this thesis;
- My colleagues (fellow GoreLab peers);
- My friends for being there when I needed a distraction or motivation;
- My brothers and sisters without whom I wouldn't have been the person that I am today;
- My parents for their unconditional love and support.

The Department of Logistics at Stellenbosch University are hereby thanked for the use of their computing facilities and office space. Any opinions or findings in this dissertation are those of the author and do not necessarily reflect the views of Stellenbosch University or PEP Stores Ltd.

Table of Contents

List of Figures

List of Tables

List of Algorithms

List of Acronyms

AD	Assign SKUs to picking lines before duplicating them
AS/RS	Automatic storage/retrieval systems
AS/R	Automatic storage/retrieval
CPO	Cube-per-order index
CSCMP	Counsil of Supply Chain Management Proffesionals
DA	Duplicate SKUs before assigning them to picking lines
DBN	Distribution list
DC	Distribution Center
EQS	Equal space
EQT	Equal time
FCFS	First come first served
FIFO	First in first out
GA	Genetic algorithm
LP	Linear programming
LSM	Living Standard Measurement
M1	Assignment method 1, cyclical assignment
M2	Assignment method 2, SLSS assignment
M3	Assignment method 3, RHLC assignment
ND	Treat non-duplicate and duplicate SKUs separately when assigning them to picking lines
NI	Nearest item heuristic
OSP	Order Sequencing Problem
PC	Original picking lines combined into a single DBN list
PS	Original picking lines treated separately
RHLS	Remaining High, Low Sequential assignment
S	Treat non-duplicate and duplicate SKUs simultaneously when assigning them to picking lines
SC	Supply chain
SCM	Supply chain management
SKU	Stock Keeping Unit
SLP	SKU Location Problem
SLSS	Set Length Subset Sequential assignment
SPLAP	SKU to Picking Line Assignment Problem
TSP	Travelling Salesman Problem
VSR	Voice Recognition System
WMS	Warehouse Management System

List of Reserved Symbols

Symbols in this dissertation conform to the following font conventions:		
$\mathcal{A}$	Symbol denoting a **set**	(Calligraphic capitals)
$\mathbf{A}$	Symbol denoting a **matrix**	(Boldface capitals)
$\underline{a}$	Symbol denoting a **vector**	(Underlined lower case letters)

Symbol	**Meaning**
b_b	Branch b in set $\mathcal{B}$
d_k	The number of duplicated SKUs on the new picking line k
d_m	The number of duplicated SKUs on the original picking line m
e_b	The SKU density of branch b, the number of SKUs to be distributed to branch b
f_{ib}	1 if SKU i is to be distributed to branch b, 0 otherwise
g_i	1 if SKU i in set $\mathcal{S}$ is not yet assigned to a picking line, 0 otherwise
l_j	The number of locations that SKU j occupy in set $\mathcal{S}$
$l_{k,j}$	The number of locations that SKU j occupy on the new picking line k
$l_{m,j}$	The number of locations that SKU j occupy on the original picking line m
p_j	The pick frequency of SKU j in set $\mathcal{S}$
$p_{k,j}$	The pick frequency of SKU j on the new picking line k
$p_{m,j}$	The pick frequency of SKU j on the original picking line m
r_m	The number of SKUs removed from the original picking line m
s_j	SKU j in set $\mathcal{S}$
$s_{k,j}$	SKU j on the new picking line k
$s_{m,j}$	SKU j on the original picking line m
x_{ij}	The number of branches to which both SKU i and j is distributed to
y_{ij}	The number of branches to which both SKU i and j is either distributed to or not distributed to
y'_{ij}	The number of branches to which either SKU i or j is distributed to
n	The number of SKUs in set $\mathcal{S}$
q	The number of branches to which the SKUs in set $\mathcal{S}$ is distributed to
C	The number of locations available on each picking line

L	The total number of picking lines to construct
M	The number of original picking lines selected
$\mathbf{F}$	Distribution matrix
$\mathbf{X}$	Similarity matrix
$\mathbf{Y}$	Inclusive similarity matrix
$\mathbf{Y}'$	Dissimilarity matrix
$\mathcal{B}$	Set of branches to which the SKUs in set $\mathcal{S}$ is to be distributed to
$\mathcal{E}$	Set containing the SKU density of the branches in set $\mathcal{B}$
$\mathcal{G}$	Set indicating whether SKU i in set $\mathcal{S}$ has been assigned to a picking line
$\mathcal{L}$	Set containing the number of locations allocated to each SKU in set $\mathcal{S}$
$\mathcal{P}$	Set containing the pick frequency of the SKUs in set $\mathcal{S}$
$\mathcal{S}$	Set of SKUs
$\mathcal{L}_k$	Set containing the number of locations allocated to each SKU in set $\mathcal{S}_k$
$\mathcal{P}_k$	Set containing the pick frequency of the SKUs in set $\mathcal{S}_k$
$\mathcal{S}_k$	Set of SKUs assigned to new picking line k
$\mathcal{B}^*$	Set of branches to which the SKUs in set $\mathcal{S}^*$ is to be distributed to
$\mathcal{L}^*$	Set containing sets of the number of locations allocated to each SKU in set $\mathcal{S}^*$
$\mathcal{P}^*$	Set containing sets of the pick frequency of the SKUs in set $\mathcal{S}^*$
$\mathcal{S}^*$	Set containing sets of SKUs

CHAPTER 1

Introduction

Contents

We find ourselves living in a fast-paced world where the speed of everything increase exponentially from year to year. We have become accustomed to getting what we want when we want it. Therefore we are increasing our demand and satisfactory standards and expecting from the world to at least meet our expectations. This behaviour is mostly driven by the advances in technology and the Internet. As a result assisting in globalization and being helped along by the impact of instantaneous communication and social networking keeping us in contact with people from different walks of life, specialising in different fields. Searching the Internet to find information regarding products and which best suit our needs and finding the most competitive price has become everyday occurrences. Today there are more than 1 billion Google searches conducted per day by more than 620 million users[1] versus 91 million Google searches per day in 2006[2]. Changes in our community and family structures also bring about change in our demands. We are constantly striving to improve what we have, driven by the goal to make our money go further, at an ever increasing pace and expecting organizations to keep up and meet our demands rather than aligning our demands with what companies have to offer as was the case up until a few years ago.

Today's supply chains are very dependent on effective logistics systems to move products from anywhere in the world to their final destination in order to meet the consumers ever changing demand. Logistics is that part of the supply chain that plans, implements and controls the efficient, effective forward and reverse flow and storage of goods, services and related information between the point of origin and the point of consumption in order to meet the customers

[1]Number reported May 2011 [18]
[2]Reported 16 April 2006 [122]

Figure 1.1: *A diagrammatical representation of the main participants within a supply chain and the product flow between them.*

requirements [132]. Logistics cost is an important part of any country's economy and therefore also of the companies which is the driving force behind an economy. When represented as a percentage to gross domestic product (GDP) logistics cost makes up less than 10% of America's GDP and more than 19% of China's [95].

Although there is pressure to reduce inventory in logistics systems, distribution centers (or warehouses) where inventory is held until needed at the retail or customer level, still play an important role in maintaining high levels of customer service.

1.1 Supply chains

The supply chain is defined by the Council for Supply Chain Management Professionals (CSCMP) as *the material and informational interchanges in the logistical process stretching from acquisition of raw materials to the delivery of finished products to the end user. All vendors, service providers and customers are links in the supply chain* [132]. The main participants in the supply chain is the suppliers, manufacturers, distributors, retail stores and eventually the consumers, as illustrated by Figure 1.1. The first participant in a supply chain is the suppliers who ultimately supply the various raw materials required to produce the products to the consumers. The raw materials are processed by the manufacturers into a form that the consumers require. This is achieved by a series of processes and by means of human activity and the implementation of technology. The distributors in turn store the finished goods until it is reassigned to the retail stores in smaller quantities. The distributors therefore also implement the process of order picking to be able to ship the correct smaller quantity of the products to the retail stores. The retail stores then sells the finished goods in even smaller quantities to the consumer, who is the end user of the finished goods.

How successful a supply chain (SC) is, depends on how well the various activities are coordinated to create value for customers while increasing the profitability of each link in the supply chain by efficiently utilising all assets. This can be achieved by implementing seven key aspects as discussed in the following paragraphs based on an article by Anderson, Britt and Favre [3].

By segmenting customers based on their particular needs, a company is equipped to develop a portfolio of services tailored to various segments. This increases the level of customer service offered to each customer. There still need to be determined up to what degree customers need to be segmented to maximise profitability and match each customer to the correct segment as this level of segmentation may vary between companies or industries. Secondly, a logistics network need to be adjusted according to the service requirements of these segments and not only maintain a monolithic approach and offer all customers the same logistical support. This increases the complexity of the logistics network and require fundamental changes in the mission, number, location and ownership structure of the warehouses.

Companies also need to respond to the market signals and align demand planning accordingly, ensuring consistent forecasts and optimal resource allocation. Over the last few years companies have moved away from factory-driven push supply chains where labor and freight cost were used to measure efficiency, to a customer-centric pull type of supply chain, measuring the number of perfect orders. By differentiating the product closer to the customer and speeding conversion between phases across the supply chain, companies strengthen their ability to react to market signals. The sources of supply need to be strategically managed in such a way to create competition between suppliers to reduce cost and maintaining a professional relationship with suppliers.

Technology need to be implemented in such a way to be able to have a holistic view of the flow of products, services and information through the supply chain in order to make well informed decisions and also reduce labor hours spent on administrative tasks. Lastly, in order to achieve a higher level of effectiveness and efficiency the goals and objectives of all the participants need to be aligned and measured throughout the process and not only for each participant individually.

Underperforming supply chains usually have unproductive assets, deliver poor service and have high variable operating cost. They tend to be narrowly focused and lack sustaining infrastructure. Their big problem may be described as being unsuccessful in the execution of various initiatives that are more often than not uncoordinated [3].

1.1.1 Warehouses and distribution centers

Warehousing is defined as that part of a firm's logistics system that stores products (raw materials, parts, goods-in-process, finished goods) at and between points of origin and point of consumption [95]. Organizations either use warehouses or distribution centers in their supply chain. The term warehouse is used when the main function of the facility is buffering and storage and their primary purpose is to maximise the usage of available storage space, while the facility is referred to as a distribution center (DC) when distribution is one of the main functions. The facility is referred to as a transshipment, cross-dock or platform center when storage hardly plays a role [26].

Placing a warehouse between a producer and customer adds a new layer of costs. Warehousing generates shorter-haul transportation routes which is more costly than longer-haul transportation. Thus there is a trade-off between warehousing and transportation.

Warehouses serves a regrouping function in a supply chain mainly focusing on matching the different rates or volume of flow between the patterns in production and consumption. Thus warehouses receives products on large scale, reorganise and repackage the products and store them in smaller quantities until they are sent to the customers. The function of a warehouse may therefore be broken down to 4 forms, namely accumulating (bringing together similar stocks from different sources), allocating (breaking larger quantities into smaller quantities), assorting (building up a variety of products for resale to customers) and sorting out (separating products into different grades and qualities desired by different target markets) [15].

Operations within a warehouse or distribution center may be classified as either inbound processes (receiving and put-away) or outbound processes (order picking, checking, packaging and shipping). These processes should take place in such a manner to achieve continuous flow through the warehouse and prevent unnecessary handling of products. By not achieving this would increase the operating cost of the facility due to the fact that each time a product is put down and picked up at a later stage takes time resulting in increased labor hours, and the smaller the handling unit, the greater the handling cost [15].

With the technological advancements over the last 20 years it has become possible to successfully model supply chains mathematically as a whole to maximise shareholder wealth and minimise cost, making supply chains more proactive rather than reactive. This includes warehouse management systems (WMS) which are able to coordinate the activities of the warehouse. By scanning products at all key decision points in the warehouse, gives total visibility of assets at all times, enabling management to make quick and accurate responses to customer demand. Scanning a package must also reflect the information about the package on the system, as this is as important as the package itself.

In the case of Pepstores Ltd. (PEP) the warehouse and DC refer to the same part of the supply chain as the activities associated with a warehouse and a DC are done at the same location, therefore for the purpose of this study there is no distinction between a warehouse or a DC.

1.1.2 Storage assignment

Storage assignment entails the assignment of products to storage locations within the DC. These locations take up space and therefore increase the operating cost of the DC due to the costs of rent, heating and/or air-conditioning and security as well as the cost of the storage equipment such as the shelves itself. The cost of receiving and storage within a DC is approximately 10% and 15% of the operating cost respectively [26]. Thus it is beneficial to use storage as efficient as possible. It is as important to determine the appropriate storage locations for each product as this contribute to how quickly and at what cost the products are later retrieved when needed. This also requires that the storage locations are to be managed in order to know at all times what storage locations are available and what are their capacity [16].

1.1.3 Routing

The routing of the (order) pickers through the warehouse to collect a number of products in specified quantities at known locations is of importance because the sequence in which the picker should visit the locations of the products that needs to be picked can minimise the travel distance significantly. Great savings is to be made on the warehouse operating cost as traveling and searching for products in a warehouse make up 70% of the time spent on order picking (as indicated in Table 1.1), which is the most expensive part of the warehouse operating cost. The order in which the products need to be picked and also the routing of the pickers through the warehouse is done by the warehouse managing system which mainly implements a *S-shape heuristic* where pickers mainly move in a S-shape curve along the pick locations skipping the aisles where nothing has to be picked [114].

1.1.4 Order picking

The order cycles may be defined as the time from when the order is placed by the customer to when the goods are received. It may also be broken down into 4 stages, namely order transmittal, order processing, order picking and assembly, and order delivery. Order transmittal is the time from when the customer places an order until the seller receives the order. Order processing

Picking activity	% time
Traveling	55
Searching	15
Extracting	10
Paperwork and other activities	20

Table 1.1: *Break down of order picking activities and the percentage of the order picking time spent on each activity in a typical warehouse [16].*

is all the activities involved from when the seller receives the order until an appropriate location (warehouse or distribution center) is authorised to fill the order. Order picking and assembly is all the activities necessary from when the location is authorised to fill the order until the goods are loaded aboard an outbound carrier. Lastly, order delivery is the time that elapse from when a transport carrier collects the shipment from the warehouse until it is received by the customer.

Order picking most often represent the best opportunity to improve the effectiveness and efficiency of the order cycle, most often without large expenditure. This is largely because order picking accounts typically for about 55% of a warehouses' operating cost [16]. A break down of the various activities involved in order picking is shown in Table 1.1. Order picking is thus considered as the highest priority area for productivity improvement in the warehouse. Traveling takes up the most time during order picking, therefore a lot of literature focus on decreasing the travel time during order picking.

The design of an order picking system in a distribution center is critical to both cost and service aspects of this relationships as selecting an appropriate picking strategy has an influence on the cost of the picking system for a given throughput requirement. It also affects the design of the aisles and storage systems, picker routing, etc. The use of automation is frequently examined as a means of reducing labor costs, but many companies continue using manual order picking due to the variability in SKU shape and size, the variability of demand, the seasonality of products, or the large investment required to automate an order picking system. The objectives of order picking is maximising throughput, minimising cost, space, response time and error-rate. Therefore the distribution center and also the supply chain's performance is directly impacted by the organisation of order-picking processes.

There are various objectives taken into consideration for warehouse design and optimisation, namely minimise the throughput time of an order, maximise the use of space, equipment and labour as well as the accessibility to all items. Factors influencing the decision of the type of picking system to implement in the distribution centers is minimum pick-rate, the requirement for a downstream sorter, blocking and workload-imbalance. Blocking implies that pickers obstruct the walkway so that other pickers can't get pass them. This results in increased waiting-time, which reduces the productivity of pickers. Workload-imbalances occur when the orders are batched in such a way that unequal workloads are assigned to pickers, which could result in orders not being fulfilled in scheduled hours of operation during a day and thus increasing operating cost due to increase in overtime. Most companies implement zone or batch picking.

There are also various strategies of manual order picking in the literature, discrete order picking, batch picking, zone picking, bucket brigade order picking and wave picking to name but a few. For discrete order picking a single picker is responsible for picking all the items in a single order during a pick-tour. It is very simple to implement, but can be very labor-intensive for medium to high throughput distributions centers and thus increasing the operating cost.

When several orders are batched or grouped together and a single picker picks all the items in the batch, it is called batch order picking. There are two types of batching, namely pick-and-sort and sort-while-pick. For pick-and-sort batching the pickers do not sort the items into customer orders and are only consolidated downstream in a manual or automated sorting system. This creates a high pick-rate (picks per unit time) for the pickers, but requires an additional downstream sorter. For sort-while-pick the pickers simultaneously pick and sort the items into customer orders. This reduces the pick-rate of the pickers but eliminates the need for a downstream sorter. Pick-and-sort and sort-while-pick batching both reduces the chances of workload-imbalances. For both cases the picker travels through the entire pick area which increase their travel time. The pickers may also block each other which can further increase the travel time.

Zone picking requires that each picker is assigned to a specific region of the storage area and is responsible for picking the items in that area only. Sequential zone picking (also known as progressive or pick-and-pass zone picking) occurs when one order at a time is picked in a single zone and then passed on to the next zone to pick all the products required for that order. This reduce the pick-rate of the pickers but also eliminates the requirement for a downstream sorter. Simultaneous zone picking (also known as synchronised zone picking) requires that all items corresponding to batched orders are picked simultaneously from all zones and then consolidated by a sorting system at the end. This increase the pick-rate of the pickers but require additional labour in the form of a downstream sorter. Both forms of zone picking eliminates congestion between pickers, but increase the chance of workload-imbalances between various zones. Thus there is a trade-off between pick-rate and the requirement for a sorting system in zone picking.

Bucket brigade picking is a self-balancing strategy and may be described as a control policy for executing discrete order picking. When the most downstream picker completes an order, he/she walk back to take over the order that the picker immediately upstream of him/her is currently picking. The latter in turn takes over from his/her predecessor and so on until the most upstream picker begins a new order. This strategy is, however, limited to applications where handing-off the items to a downstream picker is achievable. It is also most effective when the picking line is constructed as a flow line. Wave picking on the other hand is a combination of batch and zone picking where orders are to be picked in a predefined time-window, known as a wave.

1.1.5 Customer service

The overall objective of the warehouse is to deliver products to customers in such a way to minimise cost. To achieve this require that perfect orders are delivered to the warehouse's customers. A perfect order entails that orders be delivered to the right person, at the right time and in the correct quantity. Customer service is therefore a very important part of any organisation. It cost five times as much to develop a new customer as it is to retain an existing one. This creates an atmosphere where the warehouse or distribution center strives to keep customers happy and creates in the customer's mind the perception that it is easy to do business with the organization. Apart from customer service, customer satisfaction compares the customer's actual experience with the actual experience based on four factors namely time, dependability, communication and convenience.

1.2 Thesis scope and objectives

This thesis focus on the order picking system at the Durban distribution center (DC) of PEP Stores Ltd (PEP). The scope of this thesis is to develop algorithms that would assist PEP in completing the order picking process in less time with the same amount of staff by minimising the total travel distance of the pickers. PEP makes use of a picking line consisting of locations arranged around a conveyor belt that the pickers has to visit to collect the various products assigned to the branches. By minimising the number of cycles traversed in a picking line to complete the order picking, the picking time is minimised and thus branch orders are completed faster and thus decreasing the operating cost of the picking line. This is to be achieved by investigating the effect of assigning a SKU to a second bin location on a picking line, in other words duplicating the SKU.

Objective I: To describe the layout and operations of the DC so that the problem may be viewed in the broader DC context;

Objective II: To describe the order picking system in detail so that the characteristics of the problem may be understood;

Objective III: Identify constraints and make suitable assumptions so that a detailed problem may be identified and modelled;

Objective IV: Present multiple duplicating algorithms that construct picking lines to compare results in terms of the number of cycles it would take the pickers to complete the order picking;

Objective V: Discuss potential directions of future research.

1.3 Thesis layout

An introduction to supply chains in general was supplied in this chapter and the focus/scope of this study is presented. Chapter 2 gives an in depth overview of the literature associated with order picking as well as general warehouse operations. This is followed by a detailed explanation of the operations at PEP's Durban DC in Chapter 3. In Chapter 4 the problem is defined in more detail and the research questions are outlined. Chapter 5 gives some exploratory results to illustrate the effect of duplicating SKUs on picking lines which validates the importance of the study. Fifteen different SKU duplication algorithms is presented in Chapter 6, while Chapter 7 discuss the results of these algorithms on the travel distance of the pickers in the DC. In Chapter 8 the thesis is concluded based on the results discussed in the previous chapter and suggestions for future work is also suggested.

CHAPTER 2

Literature Review

Contents

Despite the fact that DCs are expensive run for any company, mostly due to the costs involved in labour, capital and information systems required, they still have an important role in the supply chain. Their main functions is to better match the supply of products with the demand of the customers, to consolidate the products and to provide value-added processing [15]. The operations within a DC may be devided into five groups, namely receiving, put-away, storage, pick and packing and shipping as illustrated in Figure 2.1.

The order picking process accounts for about 55% of the operating costs of a DC [16]. Therefore a lot of research has been performed over the last few years regarding the order picking process and related aspects. Existing research also addresses various design and operating issues with an objective to reduce order fulfillment cost or to improve overall system performance. The three process decisions considered most often are:

1. How to store the SKUs? (storage assignment strategy);

2. How to route the pickers in the DC? (routing strategy); and

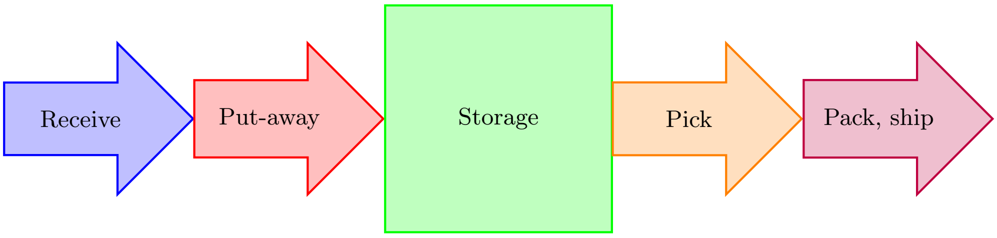


Figure 2.1: *A diagrammatical representation of the various categories of operations within a DC and the product flow between these categories [15].*

3. How to pick the SKUs? (order picking strategy).

In this chapter a broad overview is given of general DC operations, while an in-depth look is taken into previous research on the routing, storage assignment and order picking problems as well as the carousel system and the forward pick area, also known as the forward reserve area and fast pick area. The chapter is concluded with a summary of research previously performed on PEP's DC.

2.1 DC decisions and operations

There are various decisions associated with warehousing in order to keep operations running as smooth as possible and thus as efficiently as possible. These decisions and operations vary based on the type of warehouse and its main functions. Van Den Berg and Zijm [128] defines a warehousing system as the combination of equipment and operating policies used within an item picking or storage/retrieval environment. We may distinguish between three types of warehousing systems based on the level of automation, namely manual warehousing systems (or picker-to-product systems), automated warehousing systems (product-to-picker systems) and automatic warehousing systems. In a picker-to-product warehousing system pickers ride a vehicle along pick locations, this vehicle is in some cases substituted by a conveyor within the aisle which transports the picked products away from the storage or pick area (also known as a pick-to-belt system). In a product-to-picker system the picker remain in the same location and the items to be picked is transported to him by means of an electronic machine in most cases. Examples of such a system are the carousel systems and a rotary rack.

Apart from the differences in warehouse decisions and operations, there are also a difference in the complexity of warehouse decisions and operations. The two factors which mostly determine the level of complexity of the warehouse or DC were identified by Faber, De Koster and Van De Velde [43] as the number of orderlines to be processed per day and the number of SKUs being handled by the DC. This means that with an increase in the number of orderlines to be processed per day or the number of SKUs, the complexity of the warehouse or DC also increase as this require better planning. Warehouse complexity may thus be explained as the number and variety of items to be handled within the warehouse and the degree of their interaction and the technology used in order to better manage the processes which are among other determined by the position of the warehouse within the SC and its market. This in return has an effect on the daily decisions of the management team and the operations as well as which type of warehouse management system is better to implement, standard or tailor-made. The more complex the warehouse is, the more beneficial it is to implement a tailor-made system. As a rule of thumb, it is suggested that a tailor-made warehouse management system is required when there are more than 10 000 SKUs within the warehouse or if more than 10 000 orderlines are to be processed per day. Implementing a centralised inventory management system by means of a WMS increases the overall productivity and shorten the response time of a warehousing system [128].

DC operations consists of various interlinked activities that needs to be synchronised in order for the DC to operate efficiently. Bartholdi and Hackman [15, 16] gives a well-rounded and in-depth overview of the DC, from the location and layout, all activities within the DC as well as the distribution to and from the DC. The material flow is discussed as well as the various operations within a DC and the technology assisting in efficiently managing a DC. They also give a thorough discussion of the storage assignment and the required equipment to handle the stock and move it between the various locations within a DC. This is followed by a discussion of the storage of pallets based on stacking height and lane depth. A thorough discussion of the fast pick area within a DC is also given based on the estimated restocks, which SKUs and the quantity of each SKU to allocate to the fast pick area. This is followed by a self organisational order picking method namely a bucket brigade and the problem of pick-path optimisation. Heuristics are presented to generate optimised short pick-paths. They conclude with a chapter on cross-docking followed by a section on the efficiency of a DC and some different types of DCs in use across the world.

It is often found that practitioners are unaware of new research in the field of warehouse management or that the implementation of new research is not executed efficiently. In an attempt to bridge the gap between DC practitioners and academics, Gu, Goetschalckx and Mcginnis [54] compiled a comprehensive review of planning models and methods for DC operations. Van Den Berg [125] compiled a literature survey regarding methods and techniques for planning and control of warehousing systems. A hierarchy of warehousing decisions is defined which provide high quality solutions and in most cases outperform most methods used in practice. It was concluded that very few papers present optimal solutions and that the majority of the research papers make use of heuristics which is supported by the fact that most warehousing problems is NP-hard. Many publications' objective is to minimise travel time even though in practice there exist multiple objectives for each DC system like orders having to meet deadlines.

One of the factors that determines the efficiency of the DC is the operating cost [15]. This is an additional expense incurred in order to distribute products more efficiently throughout the SC. In one of the first papers on DC operations, Maserole [87] investigated the operational cost. Based on a case study where the floor space was increased by 16.6%, DC volume increased with 60% and the DC workers decreased with 60% while the working hours per person per week decreased by a further 16.6% and the annual payroll decreased by 53%, proving that there is significant savings in better utilization of the DC space and workforce. Paul and Thomas [104] used several OR techniques and common sense to improve the DC operations of an importing company in the short term in order to decrease the operating costs. The improved operations satisfied the importing company and led them to seek further long term solutions for the DC operations.

It has been established that all the parties involved in the supply chain should align all their activities for the supply chain to yield the highest possible saving on operating costs for all parties [95]. Therefore the DC network need to be investigated in order to consolidate the regional DC into a lower number of master stocking points and the subsequent phase-out of redundant or underutilised DCs without deteriorating customer service. This was modelled by Melachrinoudis and Min [92] as a mixed-integer programming model with the objective of minimising the total operating cost of all the DCs in the SC. Other non-fixed costs considered in the model was the capacity cost and the warehousing cost per unit throughput. Petersen and Aase [107], examined several picking, storing and routing policies simultaneously to determine which process decisions affect performance the most. This allow managers to determine the relative importance of the three decisions with regards to order fulfillment performance, by assisting in clarifying if a firm should implement batching of orders, optimal routing, volume-based storage, or some combination of these policies. The conclusion was made that batching has the largest impact on reducing total fulfillment time, especially for small order sizes. While volume-based and class-based storage policies also reduces picker travel time significantly.

The workforce within a DC also plays an important role in the efficiency of DC operations. Braam, Van Dormolen and Frings-Dresen [17] observed workers to evaluate the time spent on tasks, activities during the working day, postures that occurred, physiological work load, perceived exertion and recovery from work in five companies' DCs, which was classified based on their level of mechanization. More lifting of heavy objects occured in the slightly mechanised DCs than in the others. The workers in the moderately mechanised DCs had the highest estimated oxygen intake and also reported the highest perceived exertion at the end of the working day. The workers in the highly mechanised system had a shorter working day and had less problems with recovery from work than the workers of the other two systems. Poor working postures were commonly found in the highly mechanised working system. In conclusion, the highly mechanised system was considered most favorable when it concerned work load and recovery from work. Also concluding that the appliances used in the highly mechanised system, and especially the electric car, need to be designed according to ergonomic guidelines to avoid poor working postures.

Apart from the activities within a DC and the various decisions associated with the daily operations, the layout of the DC also has an influence on the efficiency. Bartholdi and Gue [13], constructed layouts for a DC that minimise the labor cost of transferring freight based on models of travel cost and three types of congestion typically experienced in cross-docking terminals. The model was implemented in the less-than-truckload trucking industry and resulted in improving productivity by more than 11% at a DC.

2.2 Storage assignment

There are various ways of storing products within a DC of which the simplest method is block storage. This method is usually implemented to stack crates of beer or softdrinks. In order to store small items, bin shelving or modular storage are implemented while large items that are stored on pallets, make use of pallet racks, gravity flow rack or mobile storage racks. The storage assignment problem entails determining which SKUs need to be located where in the storage racks or the bay locations in the picking system. The objective (in most cases) is to assign SKUs to the various storage locations in such a way that the total travel distance of the pickers or AS/R-machine (automated storage and retrieval machine), to pick all the orders, is minimised.

There may be distinguished between two types of storage, namely dedicated storage and shared storage. Dedicated storage entails that specific products are assigned to specific storage locations and that only the assigned products are allowed to be stored in those locations. This lead to popular products being stored in more convenient locations and in return the workers are able to learn which products are stored where, which speed up the product retrieval process. Dedicated storage does not use storage effieciently as on average only 50% of the storage capacity are being utilised. Shared storage on the other hand utilise storage more effieciently as a product are assigned to more than one location and when a location becomes empty it is available for reassignment even to a different product. By distributing a product over more locations the quantity per location is decreased and therefore the sooner a location is emptied the sooner it can be reassigned to another product. It is, however, more time-consuming to put new stock away as it needs to be distributed to multiple locations creating a trade-off between space and time [16]. Bartholdi also states that the average space utilisation achieved by shared storage when a product is stored in k equally sized locations, is $k/(k+1)$. This gives the graphical presentation in Figure 2.2 of the space utilisation, clearly indicating that space is utilised more efficiently by implementing shared storage rather than dedicated storage.

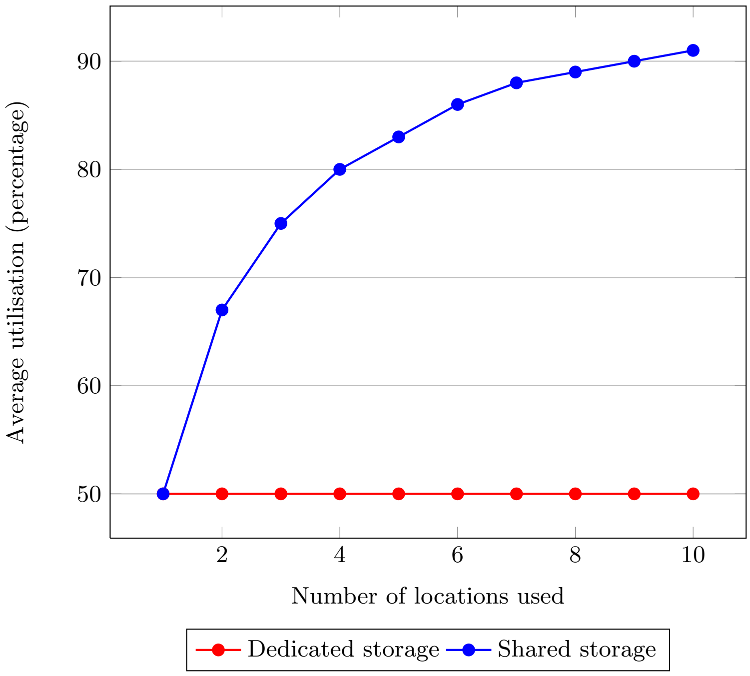


Figure 2.2: *Average utilisation of storage space in a warehouse with an increase in number of storage locations of equal size used per product for dedicated storage and shared storage strategies [16].*

From Figure 2.2 it is clear that shared storage has a higher average storage space utilisation within a DC and is thus the more effective method of the two storage policies. Malmborg [82] investigated the randomise storage (also known as shared storage) system in a multi-aisle warehouse and presented a model for developing the state probability distribution of aggregate space requirements as a function of the level of variation in item retrieval activity. This model is an extension of previous work as it accounts for space reductions associated with shared storage by estimating the average item retrieval cost. This model may also be used to analyse the tradeoffs between space requirements and the retrieval efficiency for dedicated and shared storage policies as the average retrieval costs may decline with the absolute number of storage spaces utilised in a facility, without assuming a fixed space requirement. This model gives a more rigorous estimate of space requirements for shared storage than heuristics and are difficult to solve for a large number of items. They also proved that shared storage can yield a lower average retrieval cost than dedicated storage for a 25 aisle storage system when the level of variation in retrievals demand for items is below a critical value.

The majority of previous studies uses the travel distance of the pickers as a measurement of the efficiency of the order picking process. The storage location of products have a direct effect on the travel distance of the pickers. Pan and Wu [100] developed an analytical model by capturing the operations of the pickers in a pick-and-pass order picking system as a Markov Chain to determine the estimated travel distance for a picker in a picking line. Three algorithms is also proposed to allocate items to storage locations optimally for a single picking zone, a picking line with unequal sized picking zones and a picking line with equal sized zones. Two storage policies for a unit load warehousing system was extended to a less than unit load warehousing system by Malmborg and Altassen [83], namely cube per order index item dispatching and randomised storage (closest open location item dispatching). These two dedicated storage policies was compared with regards to total item space requirements, order picking cycle times and system responsiveness. For small problems exact solutions can be generated by these methods whereas for large problems approximate results were obtained. Vickson and Lu [131] determined the optimal product location for one-dimensional storage racks, by sorting the products in descending order and to allocate to the storage racks in either a sequential or alternating manner. The optimal locations for the servers home base is either at the highest demand bin or at the median of the demand profile. For this model the assumption is made that the products are retrieved singly between successive returns of the server to the base.

The storage location assignment is furthermore explored by Ho and Sarma [59] who developed an abstract DC model for a warehousing system involving unit-load order picking and illustrate how multiple copies of items affect optimal location assignment. An optimal strategy is also presented which involves a concept called "order pressure." An optimal arrangement method for unit-load pick-orders is also presented to show how relative item probabilities affect clustering (SKUs with similar probability functions are clustered together) and distribution (product are distributed throughout the DC with localised clustering). The cube-per-order index rule was developed in order to solve the problem of locating items in a staging area to minimise the expected labour cost of order selection. Kallina and Lynn [69], discussed and summarised the background and computational steps for implementing the cube-per-order (CPO) index rule to locate items in a staging area to be picked. Some practical conclusions from applying the CPO rule to assist in DC layout planning, are also presented. Furthermore, Meller and Pazour [93], developed a heuristic to assist with the SKU assignment and allocation problem in a commonly used automated order picking system, an A-frame dispenser system.

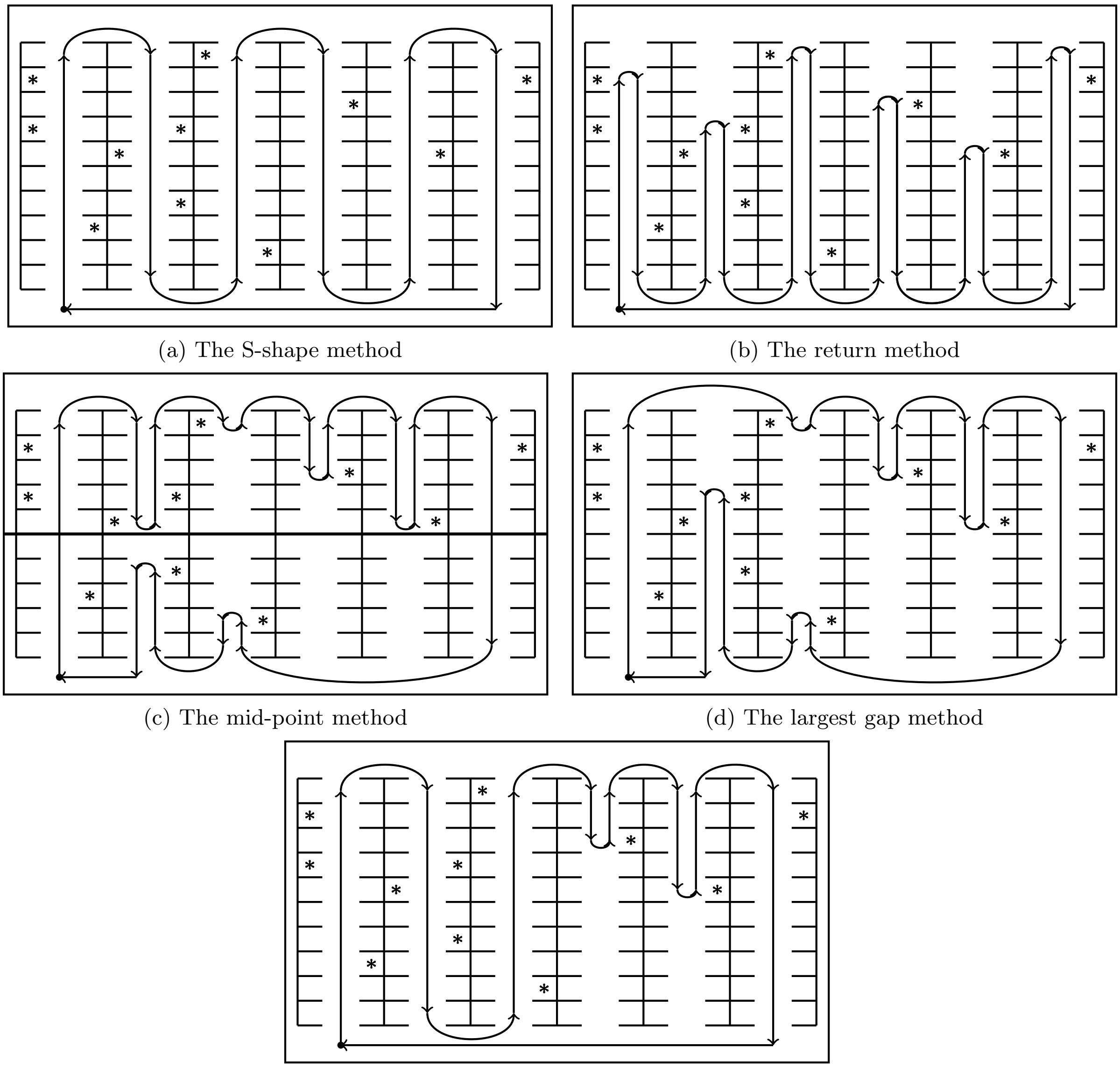
Big savings are possible by batching or clustering products with certain characteristics together, as previously mentioned. Therefor, Kim [71] developed an algorithm to cluster inventory items together in storage locations and determine the space requirement for each item, taking into consideration the inventory related cost as well as the material-handling cost per item. An improvement heuristic is also developed to reduce the total cost until no further reduction is possible. Furthermore, Malmborg, Krishnakumar and Simons [85] created a multidimensional, integrated lot size and stock allocation model to study the empirical properties of a warehousing situation with single/dual-command order-picking cycles. Gagliardi, Ruiz and Renaud [46] proposed various product location and replenishment strategies and analyzed the results for a pick-to-belt order picking system in a distribution center. It is concluded that the correct product location and replenishment strategies improve the DC's productivity by reducing the number of stock-outs in the picking area by up to 77% when compared to the actual practices of a high-throughput DC.

2.3 Routing of pickers or AS/RS machines

The routing of pickers through the warehouse to retrieve the required items from pick lists is a special case of the Traveling Salesman Problem (TSP). The TSP entails that a salesman has to visit a number of cities exactly once and have to start and end his journey at his home city. The objective is to identify the order in which the cities is to be visited that would minimise the total distance travelled when the distance between all the cities are known. This is similar to the routing of pickers with the locations of the items on the pick list being equivalent to the cities that needs to be visited. The picker routing problem is mainly solved by means of heuristics because of some of the disadvantages of optimal routing in practice. These disadvantages include the fact that optimal routing algorithms is not available for every layout. Secondly pickers tend to deviate from the designated route because the optimal routes does not seem logical to them. Furthermore, aisle congestion can be taken into account with algorithms which is not possible with a standard optimal routing algorithm [26].

There are five main routing methods of pickers through the aisles within a DC, namely the S-shape method, the return method, the mid-point method, the largest gap method and the composite method. A schematic representation of these methods are presented in Figure 2.3 with the asterisks depicting which locations needs to be visited by the picker and the same customer order is considered for all five methods. The *S-shape* (or *traversal*) *heuristic* is one of the simplest routing heuristics and entails that if an aisle contains at least one pick that aisle needs to be traversed entirely and aisles without picks are not entered (see Figure 2.3 (a)). The *return method* entails that the pickers have to enter and exit each aisle on the same end and that only the aisles containing picks are visited (see Figure 2.3 (b)). For the *mid-point method* the aisles are devided into a front half and back half. The front half is accessed from the front cross aisle and the back half are accessed from the back cross aisle. Pickers are only allowed to traverse to the back half of the aisle by either the last or the first aisle containing a pick (see Figure 2.3 (c)). Similar to the mid-point method, the fourth method is the *largest gap method* with the difference being that the picker enters the aisle as far as the largest gap within an aisle instead of the midpoint. The gap is the distance between two adjacent picks, the first pick and the front aisle or the last pick and the back aisle. A return route from both ends of the aisle is performed when the largest gap is between two adjacent picks, otherwise a return route from either the front or back aisle is used depending on which side has the largest gap (see Figure 2.3 (d)). The final method is the *composite* (or *combined*) *method* where aisles that contain picks are either entirely traversed or entered and left on the same end which is determined by means of dynamic programming for each aisle (see Figure 2.3 (e)) [26]. Hall [56] proved that the midpoint method outperforms the S-shape method when there is only a few picks per aisle required (one pick per aisle on average) while the largest gap method always outperforms the midpoint method, but it is simpler to implement the midpoint method than the largest gap method.

The routing of pickers through the storage area in a DC to retrieve products, can be compared to the routing of vehicles to a number of points. In both cases there is a depot point from where pickers or vehicles are dispatched to predefined destinations to either collect or deliver products. The scheduling of vehicles have been studied over a number of years in order to find



(a) The S-shape method

(b) The return method

(c) The mid-point method

(d) The largest gap method

(e) The composite method

Figure 2.3: *Diagrams depicting the five different routing methods of pickers between aisles for order picking within a single-block warehouse. The same order is considered for all methods. The asterisks indicate the locations that need to be picked from and the arrows depict how the picker will move between the aisles [119].*

the optimal schedule to deliver customer orders in time and in full. For example, Clarke and Wright [22] who presented an iterative procedure that enables the rapid selection of an optimal or near-optimal route for a fleet of trucks of varying capacities from a central depot to a number of delivery points.

The routing of pickers have received more attention in recent years as there have been more focus on improving the supply chain and more specifically order picking, to save on operating costs. Roodbergen and De Koster [114] presented an algorithm to find the shortest order picking tour in a DC with up to three parallel aisles which allows pickers to change aisles. It was concluded that by adding a middle aisle to the layout the order picking time can decrease significantly. They also presented a number of heuristics to determine the shortest route for pickers in a DC with two or more cross aisles [113]. The performance of the heuristics is measured against the shortest route determined by a branch-and-bound algorithm for DCs of various layouts and order sizes. The newly presented heuristics outperformed the existing ones for the majority of the instances. It is concluded that adding cross aisles to the DC decrease the order handling time by decreasing the travel distance, but adding too many cross aisles may have the opposite effect.

A genetic algorithm (GA) with Pareto elitist-based selection for the planning and scheduling of the order picking process was developed by Molnar and Lipovszki [94]. The number of pickers per shift and the best sequence to release the pick list to be retrieved from storage is determined by calculating the shortest route and creating a well-built load and evaluating a number of scenarios. More recently Theys *et al.* [124] developed a NP-hard Steiner TSP to address three questions regarding the sequencing and routing problem of pickers in a conventional multi-parallel-aisle DC system, namely

- whether the Lin-Kernighan-Helsgaun TSP heuristic [58] could obtain better solutions than existing order picking heuristics?
- whether by combining problem-specific concepts with high-quality local search features could further improve the solutions?
- whether it is necessary to use heuristics at their full complexity in order to obtain high-quality solutions?

They concluded that it is not recommended to rely only on construction heuristics to obtain high-quality solutions for the Steiner TSP and that a local search operator is able to yield better solutions.

Parallel to the routing problem of storage and retrieval machines in an automated DC is the location of the AS/RS machine when idle. Egbelu and Wu [35] used simulation to examin six dwell points, strategic placements of the AS/RS machine when idle to reduce the travel time in the DC, for dedicated and random storage policies. They concluded that the AS/RS machine needs to be positioned in such a way to minimise the expected travel time under both storage strategies. Furthermore, when the AS/RS machine is located at a point which minimise the maximum S/R travel time it dominates the remaining four dwell points in a dedicated storage policy. Dedicated storage is also superior to randomised storage for both these dwell points. Hwang and Cho [64] developed a mathematical and simulation model to reduce the travel distance of transporters in an order picking DC, taking into account DC size, rack size, number of transporters and the system performance. The results of the model were evaluated in terms of utilization of pickers, total throughput and handling time.

2.4 Order picking

In recent years a lot of focus has been placed on improving the operations within the DC in an attempt to save on operating cost within the supply chain. As a result one of the activities on which a lot of research have been conducted to have the same or improved throughput at a lower operating cost, is the order picking system within a DC. This is mainly because order picking accounts for the largest percentage of the total operating cost of a DC and also have the greatest potential for improvement. There can be distinguished between item picking and pallet-picking operations within a DC, where item picking entails that single items are picked from storage in less-than-case-loads quantities as opposed to pallet-picking where pallet loads are moved in and out [128]. Therefore De Koster, Le-Duc and Roodbergen [26] gave a literature review of work published up until 2007 on typical decision problems in the design and control of DCs with manual order picking processes. With the focus being on optimal (internal) layout design, storage assignment methods, routing methods, order batching and zoning. Of the 140 papers considered less than 30 percent focused on pickers-to-parts order picking systems. Of which most focus on random storage assignments. De Koster, Le-Duc and Roodbergen found that papers focusing on layout, batching, zoning, storage strategies and accumulation and sorting is limited but the number is increasing. Also very few authors address a combination of the decision problems as this increase the complexity of the problems. Studies on the general design procedures and global optimization models for order picking are still lacking.

The routing and storage policies in the DC and the order batching algorithms in the picking system influence the order picking system in such a way that by reducing the travel distance in the order picking system, the efficiency of the system is improved, as stated by Dukic and Oluic [33]. They also concluded that the performance is influenced by the size and layout of the DC, the size and characteristics of the orders, pickers capacity and which strategies are combined. They found that all forms of volume-based storage outperformed random storage in terms of travel distance. It is postulated that the routing policies' performance also changes when combined with volume-based storage instead of random storage and the combined and largest gap routing strategies performs better than the S-shape and return routing strategies. It was concluded that order batching has the greatest potential for reducing the travel distance in the order picking systems. For the simulated cases a modified Clarke and Wright savings algorithm was the best order batching algorithm, outperforming the seed algorithm and FCFS algorithm.

It is important to be able to determine which order picking strategy fits a specific DC best. For this reason Lin and Lu [74] proposed a two-phase computer-based procedure that can determine the order picking strategy to apply to the DC. All orders first need to be classified into five categories by using an analytic method. From there by means of computer simulation, an appropriate picking strategy that correspond to each category is generated. Petersen [106] investigated the effect that the choice of picking system have on the efficiency and cost of a picking system in a mail order company. After investigating five order picking strategies, namely discrete, batch, sequential zone, simultaneous zoning and simultaneous zone-wave picking, concluded that simultaneous zone-wave picking and batch picking are superior and that their results are not so much affected by changes in demand skewness patterns or daily order volume. It is also noted that the performance of sequential zoning and batch-zone picking deteriorates as order volume increases.

The correct decision making processes need to be followed when considering DC operations as this can influence the travel distance within a DC and thus contribute either positively or negatively to the operating cost of the DC. Manzini *et al.* [86] analyzed a picker-to-part, forward-reserve, less-than-unit-load order picking system and presented an analytical model which support the decision making process along with a multi-parametric dynamic model to estimate the travel distance in a picking cycle. They also identified which factors and combinations of factors affects the picking systems response the most. In practice distribution centers will most likely have multiple pickers assigned to the picking system at the same time. These pickers need to be coordinated in such a way to pick all orders as effectively as possible and without creating congestion within the picking system. Rubrico *et al.* [117] proposed two variations of an incremental static scheduling scheme enhanced by a local search procedure in order to schedule multiple picking agents. One is based on a steepest descent insertion method and the second based on multistage rescheduling. They concluded that on average the load-balancing algorithm (steepest descent insertion) yield the best results, while the multistage rescheduling effectively reduce the picking time when dynamism is low to moderate.

The two major contributing factors to the operating cost of the picking system is the travel distance and the pickers' behaviour. Eisenstein [36] constructed a stochastic model to compare three configurations with the objective to reduce the cost (walking distance required) of the order picking system. While Pan and Shih [99] developed a throughput model to evaluate the order picking operation performance (number of picked items per unit time) for a picking system containing multiple pickers to consider the trade-off between the picking distance and congestion delay which occur because there are multiple pickers operating on the same picking line. It was concluded that random storage policy outperformed the storage policy for picking lines where the items with the lowest frequency of occurrence on orders is stored in the most distant aisles. This is due to the fact that random storage utilise the storage area more uniformly, increasing the throughput and decreasing congestion.

Although travel distance of the picker or S/R machine is mostly considered as the measure for picking system efficiency, there are other objectives within a picking system that can measure the efficiency. An example would be the order fill rate, the rate at which orders are completed. Rim and Park [112] developed a linear programming model to assign the inventory in storage to orders in such a way to maximise the order fill rate. They showed by means of simulation that the LP approach outperforms the existing rules being followed and that the improvement ratio increases as the number of items and/or orders increase. Furthermore, Fa-liang *et al.* [44] incorporated the capacity constraints and multiple objectives of the picking system to propose a mathematical model for which an improved GA was also modelled. The GA proved to be effective, reliable and increase the effectiveness of the order picking operations in automatic DCs.

There is a lot of resemblance between various picking systems within different types of distribution centers as the end goal for all is the same, to compile customer orders from products in storage. It is, however, necessary to view case studies in order to get a more complete understanding of the implementation of models and the challenges faced when implementing these strategies. Van Oudheusden, Tzen and Ko [129] studied a person-on-board AS/R system and made recommendations based on their findings. They proposed sequencing picking tours in an aisle optimally, allowing only one operator to perform storage or order picking per aisle, create picking lists based on customers orders, redesign storage layout based on the demand frequency of each item and the closeness relationship between items and to dedicate each aisle to specific facilities. The conclusion was that picker travel time could be decreased substantially and incorporating the closeness relationship between products in the storage policy proved to be of value.

In a case study, Dekker *et al.* [25] studied the operations at Ankor's DC (a wholesaler of tools and garden equipment), in order to improve the order-picking response time. They adapted existing solutions for the storage assignment and routing problem to fit the operations for Ankor's DC. This cut the average route length in the order-picking operation by 31%. By implementing the new storage and routing strategies and the suggested improvements for the picking process, a total reduction in the number of pickers of more than 25% was realised. Caputo and Pelagagge [19] developed a decision support system and improved management criteria to more efficiently manage the operating of dispenser-based single piece automatic order picking systems (AOPS) in high-rotation high-volume DCs. This reduce the need for manual decision making and also improve the cost per picked order line.

2.4.1 Order batching or clustering

Order batching, also known as clustering of orders, entails that a number of orders are combined so that the products for those orders are retrieved from storage on the same tour in order to save travel time in the picking system. This method tend to be popular in practice even though it sometimes require having an additional sorting station further down in the picking system. Pan and Liu [98] conducted a comparative study on order batching algorithms. Sixteen algorithms were formed from four seed selection rules and four additional allocation rules. An additional SL (small and large) algorithm was also considered. They concluded that the economic convex hull method of Hwang and Lee [65] generated the most efficient batches for a small as well as a large capacity S/R machine under any storage assignment policy.

Various models and heuristics to solve the order batching problem in DCs have been developed in recent years as the focus on the order picking process increased. As the number of heuristics increased for order batching, so also the need to determine which of the heuristics performs best under which circumstances. Zoller [137] conducted extensive analysis and simulations on existing heuristics to solve the dynamic lot sizing problem and proposed an algorithm which enhances performance under conditions of erratic demand. Tang and Chew [123] again modelled an order picking system by means of a two-stage queuing system with batching and picking activities and the results was evaluated by means of simulation. They concluded that the service level may be improved by reducing the difference in the average delay time for smaller batch sizes when compared to single order picking with unlimited resources. While Gademann and Van De Velde [45] proved that the order batching problem is a NP-hard problem which is solvable in polynomial time if no batch contains more than two orders. After which the problem is modelled as a generalised set partitioning problem and presented a column generation algorithm to solve the LP relaxation. Both the branch-and-price algorithm and the approximation algorithm was tested and especially the approximation algorithm is very promising in practice because labor costs are decreased.

Order batching may be implemented in either a manual order picking system or an automated order picking system. Elsayed [39] developed four heuristic algorithms that select orders to be handled on one tour by an AS/RS machine in order to minimise the total distance traveled by the AS/RS machine in the DC. Optimal tours are found by using the traveling salesman algorithm while optimal or near-optimal solutions are found for the handling problems. No conclusion could be made as to which of the four algorithms is overall the best as the results vary when the structure of the orders or the capacity of the AS/RS machine changes. Elsayed and

Unal [41] further developed four heuristics for the order batching problem with the objective to minimise total travel time for each tour. No exact solution exists for these heuristics. They concluded that the SL (small and large) algorithm had the best performance of the presented heuristics.

The effect of batching orders on the total travel distance of the picking machines was further investigated by Hwang and Lee [65] who developed algorithms in order to minimise the total travel distance in a man-on-board automated storage and retrieval system (order picking is conducted by means of a vehicle, but a person is still required to drive the vehicle and collect the products form the storage location). The algorithms were tested by means of simulations and the results showed that they performed better than previous studies to date. In conjunction with Hwang and Lee's algorithms, an algorithm developed by Goetschalckx and Ratliff [50] that determine the optimal number of stops for a vehicle collecting the various products for a batch and the location of the stops to specify the products to be collected at each stop, can be implemented. The algorithm takes into account the trade-off between the time to stop and start the vehicle and the increased walking distance of the picker when the vehicle makes fewer stops to collect the products for the batch.

Almost all DCs vary in size, products and operating methods. Therefore Hsu, Chen and Chen [62] developed an order batching approach based on GAs for all kinds of batch structures and any kind of DC layout, that minimises the total travel distance to complete all the customer orders. Owyong and Yih [97] developed a pick-list generation algorithm, which is a heuristic control strategy aimed at reducing the order-consolidation time. An improvement algorithm is also developed to compare with the results from the basic algorithm in order to evaluate its performance. The improvement algorithm proved to outperform the basic algorithm for all the tested environments with at least 17%. Furthermore, De Koster, Van Der Poort and Wolters [27] evaluated the seed algorithm and time saving algorithm based on travel time, number of batches formed and robustness. They concluded that even simple batching methods lead to an improvement on FCFS. The seed algorithm performed best in conjunction with the S-shape routing strategy and large capacity pick devices. While the time saving algorithm performed best with a largest gap routing strategy and small pick device capacity. The seed algorithm is preferred over the time saving algorithm when CPU time is important.

2.4.2 Order batching and zoning

Another picking strategy known as zoning, entails that the storage racks are divided into zones and that one or multiple pickers are assigned to a zone and only pick the products required by the orders assigned to them in their zone. Once they have picked all the products in their zone, it is taken manually or by means of a conveyor to a sorting area where all the products from the different zones are consolidated to make up the various orders. An unbalanced workload between the various zones in the picking system can cause idle time for certain pickers when they are forced to wait for a picker upstream to finish before the new batch of orders are released. Therefore Jane and Laih [66] developed a heuristic algorithm to balance the workload among all the pickers in a zone order picking system to improve the utilization of the system and reduce the time needed to complete each order. This is a relaxation of the NP-hard homogeneous cluster model.

To better understand the effect of batching and zoning on the picking system, Parikh and Meller [102] proposed a cost model to estimate the cost of a batching and zoning pick strategy for a DC. They considered the pick-rate, picker blocking, workload-imbalance and the sorting system required. The effect of system throughput, order size, item distribution in orders and wave length on the picking strategy selection is also demonstrated. Only batch and zone picking was considered and not discrete order picking, bucket brigade or wave picking, as this is the two most frequently used picking strategies in distribution centers. Various factors, such as pick-rate, picker blocking, workload-imbalance, and sorting system requirement, that affect the batch versus zone problem were considered. A convex relationship between the sorting system cost and number of orders was found to exist. Simultaneous zone picking outperformed other strategies for large throughput systems. Yu and De Koster [136] also proposed a fast and simple approximation model for a pick-and-pass order picking system, based on queuing network theory to analyze the impact of order batching and zoning. The conclusion was that an optimal batch size always exists and the batch size does have an influence on the mean order throughput time.

2.4.3 Bucket brigade

A bucket brigade is a production line that has n workers moving among m stations, where each worker independently follows a simple rule that determines what to do next [8]. Bartholdi and Eisenstein [8] proved that a stable partition of work spontaneously emerge when workers are sequenced from slowest to fastest, independently of which stations they begin at. They also found that for typical production lines, the production rate converge to a value that is the maximum among all possible ways of organising the workers and stations. Bartholdi, Bunimovich and Eisenstein [7] furthermore described the possible asymptotic behavior of a bucket brigade production line with two or three workers each with a constant work throughput. Their work on bucket brigades was taken further by Bartholdi, Eisenstein and Foley [10], who proved the effectiveness of bucket brigades when a variability in the work content exists. Bartholdi, Eisenstein an Lim [12] also extended the bucket brigade model to be able to handle chaotic behavior in the production line when the product inter-completion times are random even though the model is deterministic.

The impact of implementing bucket brigades as a means to migrate from craft assembly to assembly lines was investigated by Bartholdi and Eisenstein [9]. This narrowed the tasks at hand for each worker and thus contributed to the faster learning and mastering of their tasks. Resulting in lower operating cost. Bartholdi, Eisenstein an Lim [11] also demonstrated how to successfully implement the bucket brigade strategy across a network of subassembly lines to balance the workload and maintain a synchronised network. The worker throughput in a bucket brigade system was studied by Armbruster, Gel and Murakami [4]. They concluded that as pickers became familiar with products on their part of the picking line, their speed increased.

2.5 Studies with combined decision problems

There is more significant savings to be gained from combining strategies than only investigating them individually. This section will focus on studies that combine the three mainstream operations within a picking system, namely storage assignment, routing and order picking.

2.5.1 Routing and storage assignment

The storage locations of products has an influence on the routing of pickers through the storage area and thus also on the travel distance of the pickers. Routing heuristics was evaluated against optimal routes in a volume-based storage environment by Petersen and Schmenner [108], as well as different methods of implementing volume-based storage and investigated the interaction between routing and storage policies under various pick list sizes and demand skewness. The performance of the routing heuristic was found to be highly dependent on the storage policy used, and to a lesser extend the size of the pick list and the degree of demand skewness. It is concluded that within-aisle storage is the best overall volume-based storage policy, and is not affected by the pick list size or demand skewness. When demand skewness increase the difference in performance between the routing policies decrease. The routing policies performance is also affected by the size of the pick list. They concluded that there is a significant difference in the mean travel distance for the routing and storage policies and that combining certain routing and storage policies yield greater savings in travel distance and therefore increasing the picking efficiency.

2.5.2 Order picking and storage assignment

The fact that the routing of pickers is a special case of the traveling salesman problem lead to Daniels, Rummel and Schantz [24] formulating several extensions of the TSP to simultaneously determine the assignment and sequencing decisions, as well as a tabu search algorithm for a DC allowing multiple storage locations per item. The main objective is to minimise the total tour cost of completing the order requirements and the results were compared to previous order picking models. These proved to be alternatives to identify low-cost location assignment and sequencing decisions. Kanet and Ramirez [70] also formulated a mixed zero-one integer program to assist a AS/RS in deciding which storage locations to retrieve an item from when given a set of discrete orders. A Monte Carlo Simulation was used by Petersen, Siu and Heiser [109] to evaluate slotting measures and storage assignment strategies in a manual bin-shelving DC. They concluded that popularity, turnover and cube-per-order index where the best strategies for slotting. They also found that golden zone storage (slotting high demand SKUs at the height between the pickers waist and shoulders) strategies generated greater savings than those ignoring the golden zone concept.

More recently Parikh and Meller [101] estimate the throughput of pickers by means of a travel-time model for an picker truck in a facility implementing a randomised storage policy (person-onboard order picking system). This model is supported by a simple, cost-based optimization model with the objective to minimise cost (of pickers, equipment and space) to identify the optimal height to store a given number of pallets. The conclusion was that with an increase in the system throughput, the lower the optimal storage level is for pallets due to travel-time issues. Also with an increase in the cost of space, the optimal storage level increase. A simulation study of a storage assignment problem (class-based storage) of a manual-pick and multi-level rack DC where the order picking performance is measured in terms of travel distance and order retrieval time was done by Chan and Chan [20]. They concluded that the DC storage system should be matched to the variety of items in the customer order for a storage assignment system to be effective.

2.5.3 Order batching and storage assignment

Order batching affects the storage location of products considerably, while storage location affects which orders are best grouped together. Ruben and Jacobs [116] proved that batching and order assignment may significantly improve order retrieval in DCs. They developed heuristics for batching under three strategies for storage assignment from the literature and applied it to walk/ride and pick systems. This lead to an increase in the number of studies being conducted on order batching within a picking system. A travel time model was developed by Chew and Tang [21] to analyze order batching and storage allocation strategies in a rectangular warehousing system, where the order picking system is modelled as a queuing system with customer batching. While Jewkes, Lee and Vickson [68] also investigated three interrelated control issues for pick-to-order picking line, namely product location, picker home base location and allocation of products. The results proved that the greedy approach is optimal for grouping products into bins and it is applicable for several alternate picking strategies. An efficient dynamic programming algorithm was also developed for fixed product locations to determine the optimal product allocation and server locations.

2.6 Carousel systems

A carousel is an automated warehousing system consisting of a large number of drawers rotating in a closed loop that are used to pick orders of items that are small, light and in high demand [77]. The drawers containing the items to be picked is moved clockwise or anti-clockwise to the location of the picker instead of the picker having to travel to the storage location. This is achieved by an automated system operated either by a person or a robot. The three types of carousel systems implemented in practice is the horisontal carousel, the vertical carousel and the rotary rack. The horisontal carousel system consists of a narrow closed loop around which a large number of carriers, driven by a chain and consisting of multiple bin locations, rotate horisontally. The vertical carousel or paternoster consists of multiple bin locations on each of the shelves that are connected at the ends to powered chains that rotate vertically. The rotary rack is a more expensive version of the horizontal carousel where each bin level is able to rotate independently, thus reducing the waiting time of the operators significantly [126]. In Figure 2.4 an example of a horisontal and vertical carousel system can be seen.

Figure 2.5 contains a schematic representation of a carousel containing a single order, where the picker is positioned at location 1 and the locations that need to be picked from is indicated by the squares. The arrows in the center of the carousel indicate the shortest distance to travel to complete the order. The carousel should move so that the picker is positioned at location 2 afterwhich the carousel should change direction and move so that the picker is positioned at location 11 and then at 8 [28].

Carousel systems are found in literature as this is a popular system to implement in a DC. However, most literature focus on bi-directional carousel systems, carousels that can rotate clockwise and anti-clockwise, as better solutions is achievable when a carousel are alowed to move in both directions. The performance of the automatic storage and retrieval system was evaluated by means of several solution procedures developed by Wen and Chang [133], that was based on the characteristics of carousel conveyors. It is essential to track the performance of the carousel system, which can be achieved by evaluating the system throughput and the picker utilization. Therefore Park, Park and Foley [103] derived expressions for the system throughput and picker

(a) An example of a horisontal carousel system (Source: [37]).

(b) An example of a vertical carousel system (Source: [38]).

Figure 2.4: *An example of a horisontal and vertical carousel system.*

utilization for a material handling system consisting of two carousels and one picker for both deterministic (when a robot is used as the picker) and exponential (when a person conducts the product picking) pick-time distributions.

It is ideal to be able to determine the shortest rotation time and number of steps of a picker before a turn in a carousel system in order for the carousel to operate more effectively. Litvak [75] was able to determined this as well as the limiting behavior and the approximate mean rotation time, for one large order with non-uniform items locations in order to achieve optimal order picking. An extensive overview and literature review of recent research on the performance evaluation and design of a carousel system by means of picking strategies for problems involving one carousel and the throughput of the systems for two carousels were presented by Litvak and Vlasiou [80]. An overview of related problems in this area is also given.

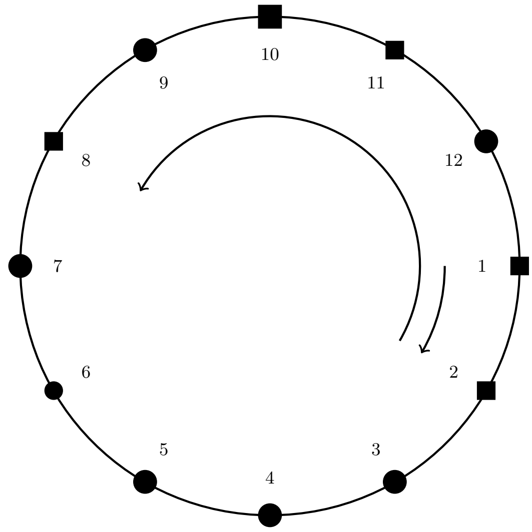


Figure 2.5: *A carousel consisting of 12 distinct locations and 1 order.*

The throughput performance of a carousel system was investigated by Ha and Hwang [55], focusing on the effect of the storage assignment policy when the server performs pickup and discharge operations. It is proved that the 2-class-based storage assignment policy significantly reduce cycle time when compared to random storage assignment policy. It is also found that the configuration of optimal shape under a 2-class-based assignment policy is dependent on system parameters such as the skewness of the inventory distribution, the shape factor and the pickup or discharge time. While Vickson and Fujimoto [130] determined the optimal storage locations in a single bi-directional carousel system. This was achieved by establishing the long-run average optimality of a simple demand rate ranking and then assigning items to bins accordingly to assist with the optimal product location for items with independent demand, as previous literature has suggested through heuristic reasoning and simulations. The travel distance to the first picked item is optimised by keeping the most frequently demanded items in close proximity. A comprehensive review of major studies on the one-dimensional storage location problem in carousel systems was given by Hassini [57]. Also presented was a general model for the one dimensional storage problem for a single carousel and other well-known combinatorial optimization problems as special cases.

As for any picking system, sequencing the orders to be picked may yield savings in travel distance. Abdel-Malek and Tang [1] modelled the stochastic cyclic sequencing problem in a drum-like storage system as a quadratic assignment problem which yielded satisfactory estimates for an optimal solution. It is also concluded that combining constructive heuristics and simple interchange procedures perform better than when working alone. An efficient dynamic programming algorithm was presented by Van den Berg [126], to sequence the picks in a set of orders on a single carousel when the sequence of orders is given. This algorithm is then simplified to a rural postman problem on a circle to be able to solve the problem optimally when the order sequence is not given. The simulation results for these algorithms showed that the average rotation time, when allowing free order sequencing, can be reduced by up to 25%.

Algorithms to determine the optimal retrieval strategies for carousel conveyors used for single and multiple orders were developed by Ghoch and Wells [49], by presenting the carousel as an alternating sequence of gaps and clusters. The method for retrieving a single order is proved to be the most efficient to date. For multiple orders a dynamic programming solution is derived under a FIFO sequencing restriction.

Litvak and Adan [77] studied the travel time needed to pick a predefined list of items when the carousel operates under the nearest item (NI) heuristic and also investigated the asymptotic behavior of the travel time when the number of items to pick tends to infinity. The NI heuristic was further applied to a carousel system by Litvak *et al.* [79]. They also determined tight upper bounds for the travel time. A simple two-moment approximation is also presented for the distribution of the travel time and the mean, variance distribution for the number of turns is also determined. The travel time required to pick n items when the carousel operates under the m-step strategy was determined by Litvak and Adan [76]. It is concluded that already for $m = 2$ is the m-step strategy is close to optimal and outperforms the nearest item heuristic and might even be easier to implement.

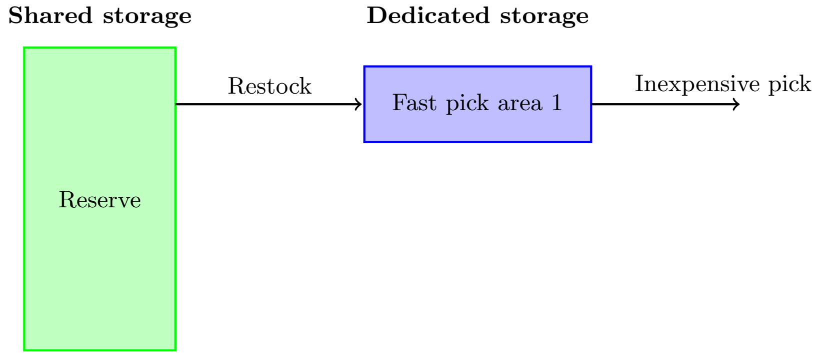
2.7 Fast pick or forward reserve area

A fast pick area (also called a forward pick or primary pick area) is a separate picking area within a DC where picks and orders are concentrated within a small space in order to improve the efficiency of the DC by improving the order picking efficiency and thus saving on the operating cost. This is accomplished by storing the majority of the fast moving SKUs in small amounts in this restriced area and therefore decreasing the unproductive traveling of the pickers. The trade-off of the fast pick area is that stock needs to be replenished from the bulk storage or reserve area. In some cases all the SKUs are stored in the fast pick area, but in some instances it is more efficient to pick large or slow-moving SKUs directly from the reserve area and allocate the space they would have occupied in the fast pick area to the fast-moving or more popular SKUs. Figure 2.6 gives a diagrammatical representation of the various ways of implementing a fast pick area and also illustrates the flow of stock within the DC. In this way the number of restocks are decreased but will require that the slow-moving items will need to be picked from the reserve area with an increase in travel time and distance. General issues that arise when considering a fast pick area is the actual size of the fast pick area, the limitations on restocking, dynamic reallocation of space or avoiding restocks by replenishing directly from receiving [14].

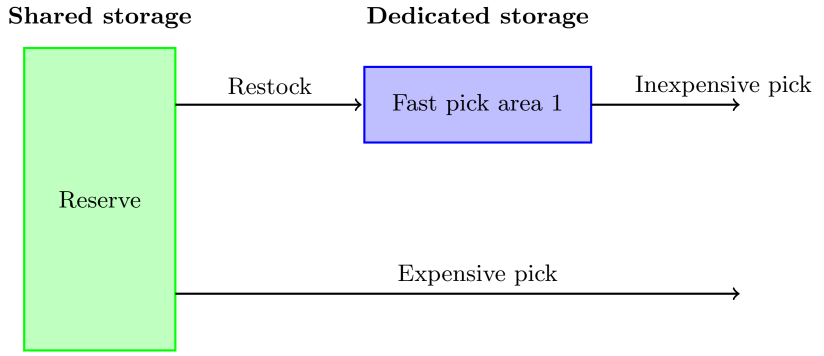
The number of SKUs to store in the fast pick area is based on the labour efficiency of each SKU, where the labour efficiency of SKU i is equal to $p_i/\sqrt{f_i}$ for which p_i is the number of picks forecast for SKU i during the planning horizon and f_i is the rate at which SKU i flow through the DC. By sorting all the SKUs from the most labour efficient to the least and then successively determining the total net cost of putting no SKUs in the fast pick area, then only the first SKU, then the first two and so on, the strategy where the net cost is a minimum is the one to implement. The net cost per SKU is determined by charging SKU i for each of the p_i picks and for each of the f_i/v_i restocks (with v_i being the volume of SKU i to be stored in the fast pick area) [15].

Bartholdi and Hackmann [15] furthermore concluded that if a SKU is to be assigned to the fast pick area, that for SKU i there must at least be a volume of $(c_r f_i)/(sp_i)$ in the fast pick area, with c_r being the cost of each restock and s the saving per pick achieved by storing a SKU in the fast pick area.

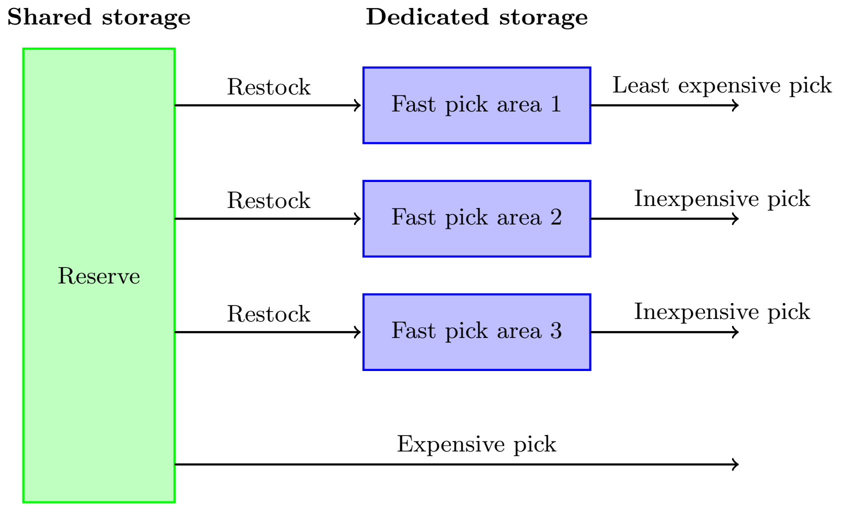
It was concluded by Bartholdi and Hackmann [14, 15] that the fast pick area is stocked in



(a) A diagrammatical representation of a DC picking all SKUs from a fast pick area and all SKUs are replenished from the reserve storage area.



(b) A diagrammatical representation of a DC picking fast moving SKUs from a fast pick area and slow moving SKUs from the reserve storage area.



(c) A diagrammatical representation of a DC with multiple fast pick areas each with different economies.

Figure 2.6: *A diagrammatical representation of the various ways of implementing a fast pick area in a DC [15].*

one of two strategies in practice, namely equal space strategy (EQS) or equal time strategy (EQT). EQS implies that the same amount of space is allocated to each SKU within the fast pick area while EQT implies that an equal time supply of each SKU is stored in the fast pick area. They also determined that of the two, EQT is implemented most often due to the belief that EQT reduce the restocks because a more popular SKU will be allocated more space in the fast pick area. They concluded that for any given set of n SKUs in the fast pick area, EQT requires the same total number of restocks ($n \sum_i f_i$) as EQS [14]. An optimal stocking strategy for small parts was constructed stating that the restocks per unit of space are equal to

$$\frac{f_i/v_i^*}{v_i^*} = \left(\sum^n \sqrt{f_j} \right)^2 \tag{2.1}$$

if the quantity of SKU i to be stored in the fast pick area is

$$v_i^* = \frac{\sqrt{f_i}}{\sum_{j=1}^n \sqrt{f_j}}. \tag{2.2}$$

This proved that both EQS and EQT are suboptimal when compared with the optimal strategy, meaning that too much space is allocated to some SKUs while too little is allocated to other SKUs. The conclusion is that the optimal space allocation strategy will never be smaller than the EQS and never greater than EQT, as presented in Figure 2.7. When implementing either the optimal allocation or EQT strategies it is neccessary to forecast the flow of each SKU over the planning period.

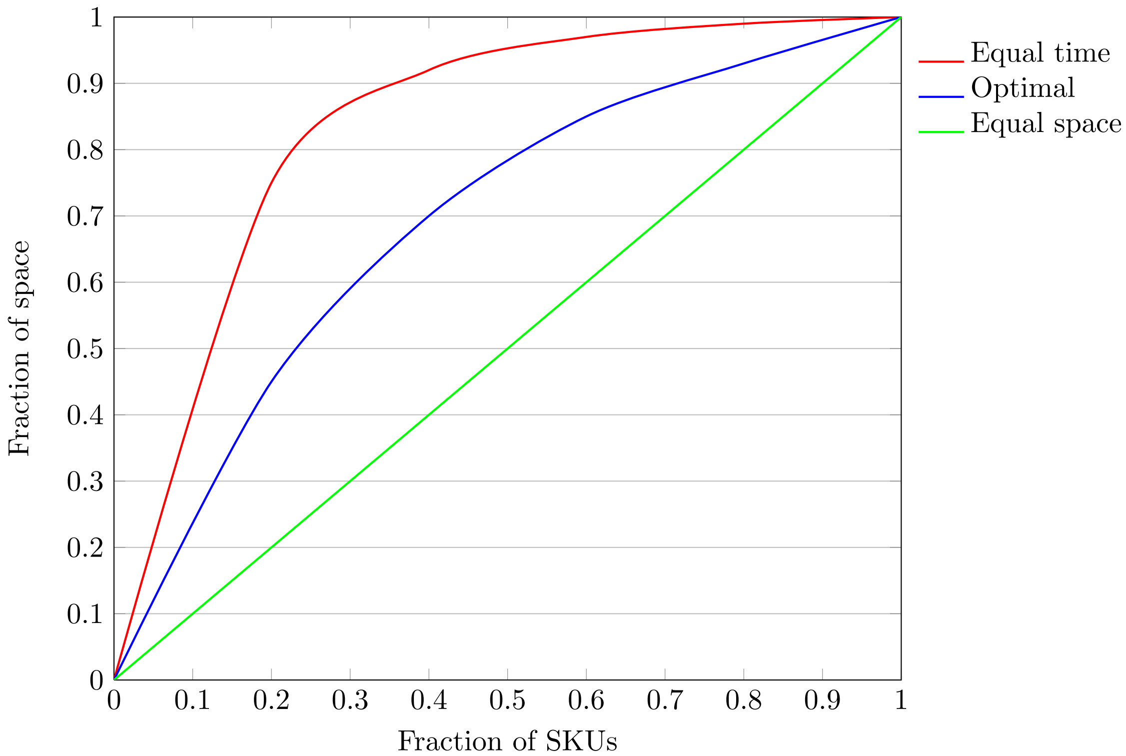


Figure 2.7: *The curves display the cumulative fraction of space consumed by the equal space, equal time and optimal space allocation strategies [14].*

The influence of adding buffers in a fast-pick area of a DC so that pickers are able to work ahead, was evaluated by heuristics proposed by Kong and Masel [72]. The heuristics assisted with assigning SKUs to the zones as well as sequencing the orders for picking. They concluded that the throughput per picker is increased and therefore outperforms the systems with random assignments or wave picking.

After studying a warehouse with 200 products and 800 storage locations, Van Den Berg and Zijm [128] presented a forward/reserve policy that reduced the order pick time by more than 40%. They also discussed the class-allocation method, which in a DC of a specific motor company, reduced the travel time by 10% when compared to the current four class-based strategy. They concluded that sophisticated class-allocation leads to higher overall service levels in the DC since storage space is better utilised and that the higher service level and shorter response time may lead to additional savings further down the SC. They also noted that further reductions in inventory levels and improvements in response time is possible with a better understanding of the warehousing system and its key factors contributing to control and design of the warehouse as most researchers assume the storage and material handling as given.

2.8 Previous work on PEP's picking line

A few studies have been performed in recent years based on operations at PEP's DC's. One of the first studies was conducted by Scott [119], who developed an exact formulation to minimise the total travel distance of the pickers in the PEP DC, as well as an alternative meta-heuristic approach. The meta-heuristic approach determine the location of products on the picking line to minimise overall travel distance for the pickers. Furthermore, the sequence in which the customer orders should be picked are also determined. The tabu search approach outperformed the random allocation of products. The methods improved the results, but for the large-scale data sets the improvement in the efficiency of the picking line is less significant. It was concluded that there is little to be gained in travel distance in the picking lines by changing the allocation of products and the sequence of the orders in a single aisle picking line. It is determined that the focus should be on the construction of a picking line, the decisions involved in assigning products to the line and determining which customer orders to be assigned to which picking line.

At the Kuilsriver DC, pallet-loads of stock are moved between the storage area and picking area by means of cranes. Matthews [89, 91] investigated the management of the crane movements by focusing on the effects of integrating the different types of jobs performed by the cranes between the storage racks and picking lines. A model was presented as a variation of the TSP. A re-optimisation approach assisted in handling the dynamic element of the problem. Dynamic re-optimisation was incorporated into the problem by means of a tabu-search and ant colony meta-heuristics and combinations thereof as the solution time needed for the exact formulation was too long. It was shown that the ant colony algorithms were not as effective as the tabu search and implementing a combination of a tabu search and ant colony methods in order to determine the crane movements within the Kuilsriver DC yields better re-optimisation results and that the proposed model outperforms the philosophy that was implemented by PEP at the time.

The Durban DC's picking area consists of five 112-location picking lines (or ten 56-location picking lines) and 1 permanent picking line. The permanent picking line is used to pick the non-seasonal items that are distributed to the various branches across the country all year round and thus need to be in stock throughout the year. These non-seasonal items typically include schoolwear and underwear. Samuels [118] considered the assignment of products to the permanent picking line in order to minimise the number of replenishments and assist in minimising the travel distance for the pickers. The maximisation of the TSP was used to assign products to bays along the picking line. From the results obtained from heuristics solving the TSP it is concluded that there is a trade-off between the number of replenishments and the travel distance of the pickers. It is also concluded that when SKUs are paired up with each other a level of flexibility is lost when assigning SKUs to the flow-racks. There also exists a relation between the variation in bay sizes and replenishment and the total distance traveled to pick all the orders.

PEPs order picking process at their Durban DC implements wave picking and the picking line consists of pallet loads of SKUs arranged around a conveyor belt while pickers move in a clockwise direction around the conveyor belt in order to pick the various orders. The order picking system implemented by PEP therefore is similar to the uni-directional carousel system found in literature. De Villiers, Matthews and Visagie [30] categorised the order picking process of PEP at the Durban DC into three decision tiers, namely the SKU to Picking Line Assignment Problem (SPLAP), the SKU Location Problem (SLP) and the Order Sequencing Problem (OSP). The decisions associated with these three tiers are made sequentially and therefore each problem relies on the information generated by the predecessing tier. De Villiers [28] addressed the OSP first and an exact formulation as well as a number of heuristics and metaheuristics are presented to solve this decision tier. A relaxation of the problem reduced the size. The OSP was further solved by means of a number of greedy tour construction heuristics, a scope and ranking algorithm, methods based on awarding starting locations with respect to preference ratios and a modified assignment approach. With the generalised extremal optimisation approach delivering the best solution quality when the results and computational times are compared with the data provided by PEP. For the SLP little or no improvement was achieved by all the methods used to determine the SKU locations within the picking line. This included two methods from literature, an ant colony system that maximised the number of orders in common between adjacent SKUs and two novel heuristic clustering algorithms of which the first calculates the distance between two clusters as the set of orders that have to collect all the SKUs in both clusters and the second is based upon the frequency of the SKUs within the cluster. A number of agglomerative clustering algorithms were also used from which dendrograms could be constructed. For the SPLAP a possible exact formulation is presented as well as a nearest neighbour search which was initially used to construct new picking lines based on all data sets. A tabu search was also conducted where the waves of two or three picking lines were altered. This resulted in significant savings for large data sets.

Matthews [90] established a tight lower bound for the OSP by implementing a concept of maximal cut that was transformed into a feasible solution within 1 pick cycle of the lower bound which proved to be dynamic and robust for use in practice. Four variations of a greedy heuristic and two metaheuristic methods were developed which were able to solve the problem in shorter times as the exact formulation's computational time were to large for real life data sets. The SLP was solved by means of an ant colony approach. In order to cluster the SKUs together on a picking line, four variations of hierarchical clustering algorithms were developed while three metaheuristic methods were developed to sequence these clusters. These methods all outperformed known methods for assigning locations to SKUs on a carousel. By means of simulation models all methods proved to be applicable in practice and the proposed methods for the SLP and OSP outperformed the current approaches implemented by PEP. The conclusion was also made that the OSP is of higher importance to PEP in comparison to the SLP as very limited savings can be achieved when solving the SLP.

De Villiers and Visagie [29] presented two classes of heuristics in order to solve the problem of determining in which sequence the orders should be picked such that the travel distance of the pickers are minimised. The first class is based on a generalised assignment approach while the second utilises preference ratios of which a variation of the latter yielded the best results. De Villiers, Matthews and Visagie [30] presented eight tour construction heuristics that was implemented for a order picking system operating in unidirectional picking lines. From this the tour construction starting position (TCS) and the tour construction ending position (TCE) heuristics was developed to sequence orders in a picking line. It was found that a TCE heuristic with adaptations obtained the best solution quality when compared and tested with real life data sets.

Placing products in storage and restocking a product takes up time, it also increase the product handling which increase the operating cost. Therefore Matthee [88] investigated the re-stocking of SKUs on a picking line with the objective being to minimise the number of restocks by assigning different size bins to each SKU on the picking line, which operates as a forward reserve or fast pick area described in literature. A variation of the space allocation model are presented and template allocations are incorporated. By using bins of different sizes on a picking line implementing a variation of a kanban system in PEPs Durban DC, approximately 13% savings may be achieved on the total number of restocks when compares to the current number of restocks on the picking line. A kanban system assists in achieving just in time production by managing each process so that the correct number and type of components required by the process are pulled at just the right time. A single card kanban system is defined as when *parts are produced and bought according to a daily schedule, and deliveries to the user are controlled by a "conveyancing" (withdrawal) kanban* [2]. A total saving of 25% can be achieved in the number of restocks when two consecutive locations are considered for a SKU.

2.9 Chapter summary

A lot of research focused on warehousing and order picking, mostly due to its high cost contribution to the overall expenditure by a company. The majority of papers presents improving heuristics and very few papers present exact solutions due to the complexity and solving time. Travel time is mostly used as a measure for the effectiveness in order picking systems. By minimising travel distance the efficiency of the pick system is improved [33]. Another measure is the order throughput for total DC operations.

While the majority of the papers focus on one of the three main decision strategies in a DC [26], it has been proven that combining for example routing and storage policies, yield bigger savings on travel distance (by the picker) than when these decisions are treated individually [108]. By combining batching and storage assignment, the order retrieval in a DC may be improved significantly [116].

Volume and class-based storage also reduce picker travel time significantly [107]. Storage policy effectiveness can be increased by applying the golden-zone storage policy. 2-Class-based storage outperforms random storage strategies in carousel systems. Shape configuration for 2-class-based storage is affected by the skewness of inventory distribution, the shape factor and pickup and discharge time [55]. Keeping the most frequently picked items in close proximity saves travel time in carousel systems [130], but can also be applied outside of carousel systems. The difference between the performance of various routing policies is decreased with an increase in demand skewness. Their performance is also affected by the size of the pick-lists [108].

The batching of orders and zone picking is the most popular picking strategies used in DCs [102].

Batching of orders proved to have the largest impact on order fulfillment time [33, 107]. Even simple batching methods improves results when compared with results obtained from FCFS systems [27]. An optimal batch size does exists for order batching and the batch size does not have an influence on the mean order throughput time [136]. Some studies proved that the greedy heuristic is optimal for grouping orders in bins [68]. In some cases the batching heuristics vary too much when the order structures and picker/machine capacity vary to determine which is the better batching heuristics [41]. Simultaneous zone-wave picking and batch picking are superior to other picking strategies and their results are not so much affected by changes in demand skewness or daily order volume [106]. The performance of sequential zoning and batch-zone picking deteriorates as order volume increase [106].

DCs that uses a seperate picking line system from the storage racks in which pickers themselves does not retrieve SKUs from the storage racks, as is the case at the PEP DCs are defined in literature as forward reserve picking or fast pick areas. PEP currently make use of the equal space forward pick area where all SKUs are assigned to the fast pick area and operate on the same principles as a uni-directional carousel. Very few studies have focussed on uni-directional carousel systems where the carousel is allowed to rotate only in one direction, either clockwise or anti-clockwise. The majority of literature focus on bi-directional carousel systems that can rotate both clockwise and anti-clockwise.

CHAPTER 3

PEP overview

Contents

PEP is a devision of Pepkor Retail Limited and is the biggest single brand store network in Southern Africa and also owns and runs the largest clothing factory in Southern Africa. It was founded in 1965 and has since grown to more than 1 400 stores in 9 African countries (there is a PEP store in almost every town and village in South Africa) with revenues of R10 billion and employing about 14 000 people [67]. Based on the principle to supply quality merchandise at affordable prices to everyone in South Africa, they describe themselves as a cash-based, low-margin, high-volume clothing retail store. This has lead to their success and enabled them to expand their portfolio to include cell phones, airtime, prepaid electricity and insurance. They maintain a very innovative business model with the main objective to constantly seek to bring additional value to their LSM 2–6 target market[1]. PEP's target market is therefore mostly people falling in the low to middle income group of the population.

PEP sells more than 400 million products through 220 million customer transactions per year and therefore requires efficient and accurate processes at its distribution centers and throughout the distribution network. In order to shorten lead times, respond swiftly to customers' changing needs and reduce prices further, PEP continue to invest in its supply chain, planning and merchandising departments, which has become a contributing factor to their success.

PEP's distribution network consists of 3 distribution centers and 17 hubs servicing over 1400 branches across South Africa. The distribution between the hubs and the branches is managed by SCM360[2]. Of the three distribution centers, namely Durban, Johannesburg and Kuilsriver,

[1] The South African Advertising Research Foundation's (SAARF) Living Standard Measurement (LSM) divides the population into 10 LSM groups with 1 being the lowest and 10 the highest. Grouping is based on 29 factors including degree of urbanization, ownership of cars and of major appliances [121].

[2] SCM360 is a division of Pepkor Retail Ltd. who manages all PEP's distribution.

Figure 3.1: *Aerial view of PEP's Durban distribution center, covering 63 000 m^2 (Source: [34]).*

Durban is the largest and handles the majority of the volume of all the stock. This is due to its easy access to the East, especially China, via the Durban harbor which handles more than 60% of South Africa's imports and the fact that about two thirds of the products are imported. An aerial view of the Durban DC can be seen in Figure 3.1. It is also used as a hub to service the branches in the vicinity directly. The Durban DC covers 63 000 m^2 of which 1 700 m^2 is used as office space that serves the needs of the distribution, warehouse management and general material handling employees. In this chapter an overview of the operations within the Durban DC is given.

3.1 Distribution network

Distribution accounts for the third largest cost in PEP's financial statements. Figure 3.2 illustrates PEP's distribution network. Products are received by the three DCs from the various suppliers and factories in bulk. Smaller quantities is then distributed to the hubs, who then distribute the products to the various branches. As PEP owns the factories providing them with some of the desired products and all the hubs and stores belong to PEP as well, it eases the synchronization and coordination of product flow through the supply chain, resulting in a more efficient supply chain as discussed in Chapter 1.

Annually more than 2.5 million cartons are distributed from the Durban distribution center to the more than 1400 stores in South Africa. Trucks leaving the three DCs or 17 hubs for distribution travel far enough to circumnavigate the globe 280 times annually [6] as the average distance between the DCs and retail outlets are 1 300 km. PEP's DCs strive to reduce stock on hand and improve service levels in stores, by buffering more stock. The Durban DC is also believed to be one of the largest single-roof distribution facilities in Africa [105].

3.2 Planning

The Durban DC serves all branches across South Africa, which is located in various areas and communities and therefore targeting different market segments. It is therefore necessary to differentiate between stores to provide varying selections of products while still maintaining the base range of seasonal and non-seasonal products across all the stores.

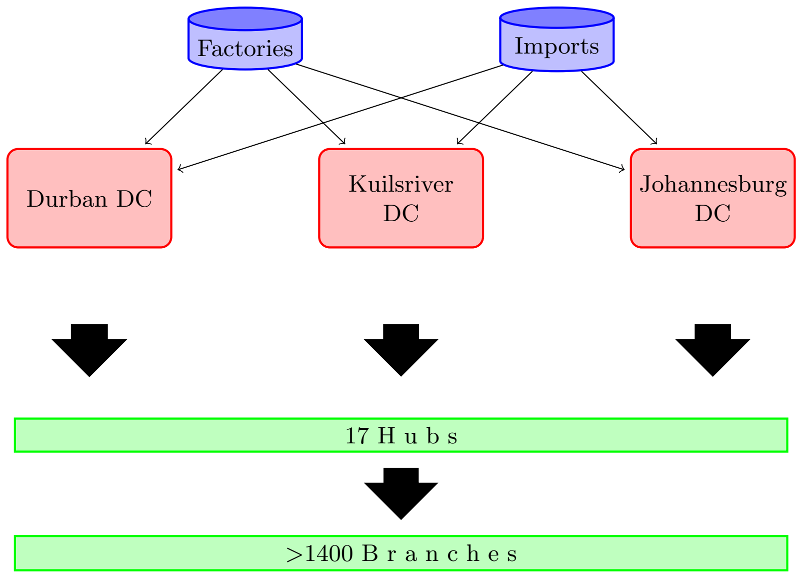


Figure 3.2: *A diagrammatical representation of PEP's distribution network.*

PEP minimise managerial functions at store level, therefore the planning of SKUs to be distributed to branches is performed at central office by the planning department for all stores. This ensures uniformity amongst the product selection available as the same team of people decide on the products and target market of all the stores. An inventory-driven culture is maintained, not a receipt-driven planning culture. In other words, planning is conducted on the basis of distributing each SKU to its various branches, rather than distributing all the SKUs to specific branches at a time. This enables PEP to pick all the branch requirements for the SKU and ship it to all the relevant branches in one operation. Approximately 50% of PEP's stock is for replenishment, in other words stock that are sold throughout the year that are not subject to seasonal change or the latest fashion trends for example linen, underwear and school wear. The remaining 50% of stock is seasonal, in other words it is only sold for a single season after which it is discontinued and replaced by new products as fashion trends change from season to season [28].

Each branch's SKU requirements are forecasted on a weekly basis and branch orders are planned accordingly, with the goal to minimise inventory levels in stores. Planners review plans on a cyclical basis to monitor buying patterns and adjust the strategy as required, to meet the demands of the customer base of each of the branches. The DCs manage a stock-older-than-one-year level of 3% [67].

3.3 Warehouse operations

Operations at the Durban DC may be divided into three main categories, namely receiving and storage, order picking and distribution. Figure 3.3 illustrates the layout of the Durban DC.

Merchandise is received at the receiving area, where trucks pull into the docking area and gets unloaded. Before sending newly arrived stock to the various storage areas to clear up floor space for the next load of stock, quality control is performed to do a quantity check on the merchandise and to check that the stock is of a desired standard. All incoming cartons are registered on the

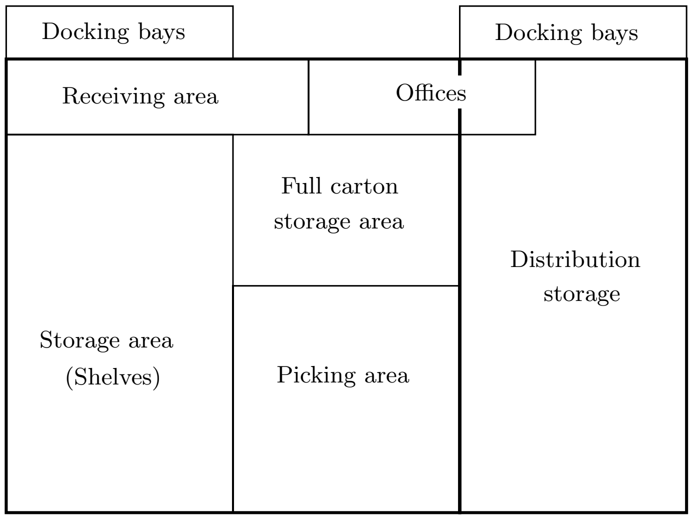


Figure 3.3: *A diagram representing the internal layout of PEP's Durban distribution center.*

warehouse managing system (WMS) by scanning them, to provide a clear picture of stock on hand at all times.

PEP uses a volume-based storage policy, where SKUs with the largest demand is located close to the picking lines, which reduces travel time and locating time when stock has to be retrieved for picking. The cross aisle between the storage racks and pallets of received stock waiting to be loaded onto forklifts and placed in the designated storage location, may be seen in Figure 3.4. In Figure 3.5 the side view of the storage racks is visible. Each storage rack is 5 levels high, with each level high enough for a full pallet to easily fit into the racks. Aside from the storage racks, there is also floor storage available that is predominantly used to store cartons that will be shipped as full cartons and does not require picking. In the full carton storage area the full pallets are often stacked on-top of each other if the cartons are strong enough to carry the weight.

Forklifts and pump trolleys are used to move stock between storage, locations on the distribution center floor, as well as constructing picking lines, replenishing low stock of functioning picking lines and removing leftover stock from picking lines after each group of branch orders have been completed. Specialised forklifts are required to place and remove pallets in the storage racks due to the height and width of the storage racks and the aisles in between the racks. The narrow aisles between the storage racks also increase the space utilization in the warehouse.

The next group of operations in the warehouse to follow receiving and storage is order picking. A distribution list (DBN) is defined as the quantities of specific SKUs destined for all branches. Effective DBN management, the management of various activities associated with the DBN cycle, is crucial to the warehouse operations and can result in gains in the overall performance of the distribution center. Order picking entails consolidating the DBNs received, so that it meet the specifications of all the branches. In other words the SKUs need to be unpacked manually from the larger cartons to smaller cartons as specified on the DBN. All the DBNs are grouped into a wave and the branch order are then determined from the present DBNs. A branch order is thus wave specific. It is a very time consuming and labour intensive exercise and therefore also the highest cost operation in the distribution center.

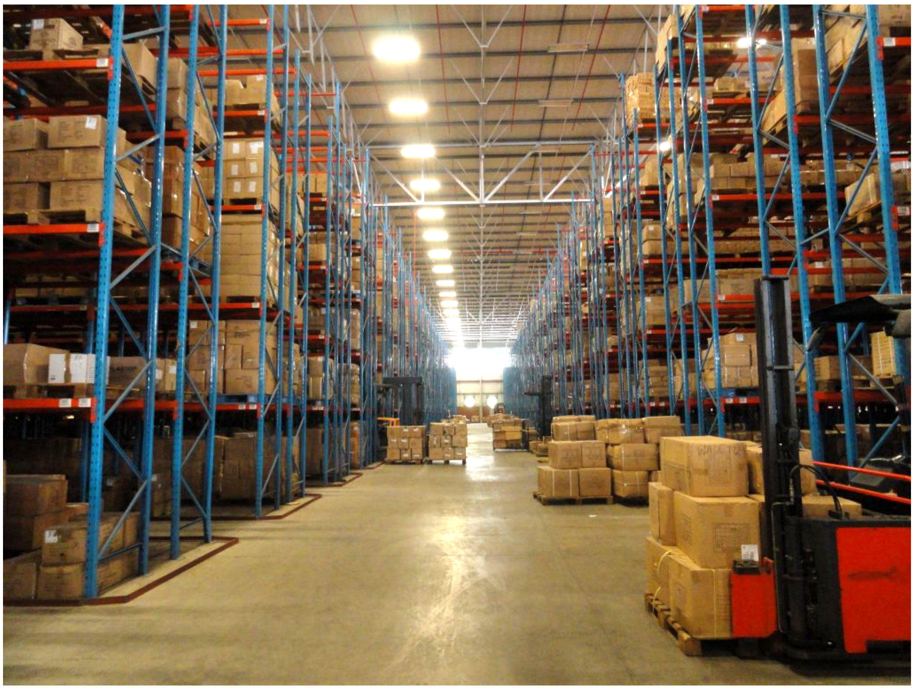


Figure 3.4: *A cross-aisle view of the storage racks at Durban distribution center.*

Figure 3.5: *A side view of storage racks at Durban distribution center.*

Figure 3.6: *A picker wearing a headset used to implement eyes- and hands-free solution and pushing a trolley loaded with cartons, used for picking branch orders.*

PEP furthermore uses an end-to-end speech recognition solution that allows speech-directed interaction between the mobile worker and the host system at the three distribution centers. The headset used by the pickers may be seen in Figure 3.6. The trolley used to assist in the picking process is also visible in this figure. This system guides workers step-by-step through their daily tasks, improving speed, accuracy, performance and productivity for order preparation, picking, etc. By implementing the eyes- and hands-free solution, PEP has increased productivity by 15% and improved accuracy by 19% [110].

The complete orders also go through a quality control station (see Figure 3.7) where randomly selected cartons are checked to determine the accuracy of the pickers by checking the SKUs and quantities thereof in the selected cartons. Only then do the cartons gets sealed for distribution (see Figure 3.8). Once all the branch orders have been completed, the cartons are sorted by destination in the distribution storage area until the scheduled truck arrives to collect all the cartons for the hub. The conveyor belts coming from the picking area to the distribution storage area may be seen in Figures 3.9 and 3.10.

3.4 Picking line process

The picking lines are the main bottle neck in the DC as the picking process is time consuming and the smaller the handling unit of the stock becomes the longer it takes to handle. It is for instance a lot faster to move a pallet of school shirts than single units at a time.

PEP implements a system of wave picking, which is a combination of order batching and zoning, where each picker is responsible for SKUs in his/her zone for numerous orders [107]. The

Figure 3.7: *The quality control station of a picking line at Durban distribution center.*

Figure 3.8: *Employees sealing cartons after picking of branch orders at Durban distribution center.*

Figure 3.9: *Conveyor belts coming from the picking area brings the cartons ready for distribution to hubs, to the distribution side of the DC to store in the designated floorspace before being loaded onto trucks (Source: [23]).*

Figure 3.10: *Conveyor belts coming from the picking area brings the cartons ready for distribution to hubs, to the distribution side of the DC to store in the designated floorspace before being loaded onto trucks (Source: [23]).*

branch orders are batched by SKU and not by branch. This type of order picking is especially beneficial to larger distribution centers, even though it usually requires secondary operations to consolidate the orders. PEP elliminates the need for consolidation of orders by assigning a single branch order to a single picker and therefore the product does not get mixed in a carton with those products assigned to another branch. It is also effective as they have a fixed set of branches to which all the SKUs gets distributed to, and can easily plan waves on a SKU level so that all the branches receive the same stock at more or less the same time. Thus picking lines are constructed in such a way that they contain a set of SKUs associated with a wave and all branch orders relevant to the SKUs on the picking line has to be completed before new SKUs are allowed to be placed on the picking line.

The picking lines are constructed on one level and the structure consists of a conveyor belt of approximately 75 meters with 56 locations on each side (see Figure 3.11). The conveyor belt contains a gate (that can be opened or closed as required) which allows for the picking lines to be treated as two picking lines with a capacity of 56 locations each or a single picking line with 112 locations. The shorter picking lines takes less time to construct and decrease the distance per cycles as well as the time for the pickers to walk a cycle. Ultimately saving on operating time and labour cost. The width of each location is equal to one pallet (1m $\times$ 1.2m) and is long enough to fit five pallets behind each other. Each location therefore contain a different set of identical SKUs which remain in the allocated locations until all the orders for the specific picking line has been completed by the pickers. Empty locations next to the conveyor may be seen in Figures 3.12. The Durban DC have 5 picking lines of 112 locations (10 picking lines with a capacity of 56 locations) and another permanent picking line (with a capacity of 288 locations, containing flow racks instead of one level of locations on the floor) designated to the products that need to be distributed to the branches throughout the year, in other words non-seasonal or replenishment items (see Figure 3.14). Picking line managers thus have the option to plan picking lines containing up to 112 SKUs or picking lines containing 56 SKUs, the latter is currently the more popular option of the two.

The first-in-first-out (FIFO) system is implemented with regards to stock rotation in PEP's DCs. Therefore the first SKUs for which the actual product and its DBN from the planning department is received at the DC will be processed first.

Once the DBNs are received from the planning department, the SKUs are split between picking lines so that the number of picks is evenly distributed and the picking line can be constructed accordingly. SKUs are arranged in such a manner around the picking line so that the SKUs with the highest pick frequencies are evenly distributed and thus placing SKUs with a high pick frequency next to one with a low pick frequency and in an effort of preventing congestion between pickers. The SKUs need to be removed from the storage racks and placed in the locations of the corresponding picking line with the use of forklifts and pump trolleys. Empty picking lines are visible in Figures 3.12. The picking line in Figure 3.13 has stock that have been retrieved from the storage racks and placed in the locations of the picking line.

The PEP distribution center operates 24 hours a day, therefore the picking line is constructed during the night shift to be ready when the pickers arrive in the morning. This also increase effectivity as there would have been idle pickers that would be waiting for the construction of the picking line to finish. The picking lines need to contain enough units of each SKU so that all the branch orders can be picked from the picking lines to be distributed to the branches. A team of eight pickers will typically be assigned to each picking line to complete the branch orders.

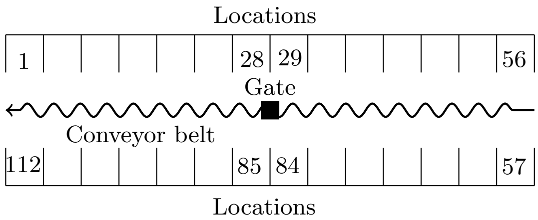


Figure 3.11: *A schematic representation the picking line structure, a conveyor belt with locations on both sides and a gate in the middle.*

Figure 3.12: *An empty picking line at the Durban distribution center.*

Figure 3.13: *A full picking line at the Durban distribution center.*

Figure 3.14: *The permanent picking line at the Durban distribution center.*

Figure 3.15: *A picker busy picking a branch order at the Durban distribution centre.*

Pickers have small trolleys on which they place empty cartons to fill with SKUs as instructed by the voice recognition system (VSR). There is enough space for two trolleys to fit in next to each other in the area between the locations and the conveyor belt. To prevent congestion, pickers are only allowed to walk clockwise around the conveyor belt. Thus the additional space makes provision for pickers to overtake each other. Figure 3.15 contains a photo of a picker walking with her trolley around the conveyor belt.

Branch orders are picked in serial, meaning that each picker is responsible for completing one branch order at a time, as opposed to parallel picking where multiple pickers all pick the same branch order. It is allowed that the SKUs relevant to a branch is spread across more than one picking line, but the picker will only pick the SKUs for the branch on his picking line and will not leave his picking line.

Each picker has to record their voice on the voice recognition system when they are employed by PEP. This recording is used along side voice recognition software to identify the picker when they sign in at the beginning of a shift. If the system does not match the picker with his voice, it will not continue giving any instructions. Once they have signed in and the software has identified the picker, the system will select the first branch order available at the top of the list and direct the picker to the nearest SKU for that branch order. A label is also printed out that is placed on the carton, which has a unique barcode on to help identify where the carton needs to be distributed to and what is the contents of the carton. When the picker arrives at the correct location, a second check code at the locations is read to the software to verify that the picker is indeed at the correct location. To prevent that pickers remember this second check code of the location and repeat it to the system as soon as they are directed to the location, these codes are updated on a regular basis. When the correct code is read back by the picker, the system instructs them to the quantity of that SKU that they need to pick for the branch. After the correct quantity is picked and placed in the carton on the trolley and reported back to the VSR, the system will guide the picker to the next SKU on the branch order until the branch order is complete.

If a carton is full but there is more SKUs that need to be picked for the relevant branch, the picker can feedback to the VSR that the carton is full. In this case the full carton is placed on the conveyor belt which takes it to the quality control station and the sealing area, from where it is taken to the distribution storage area. The system then prints out a new label to be placed on the new carton containing a barcode. This barcode is scanned at dispatch where another sticker is printed out containing the detail of where the carton is heading and the contents thereof. When all the items are picked for a branch order the system will inform the picker and the picker will place the carton on the conveyor belt. The picker then collects a new carton for the next branch order. As cartons are emptied during picking, pickers will collect them from the locations to use for packing the branch orders.

All the SKUs is further classified as either a type A, B or C pick. A type A pick can be picked using one hand. A type B pick is a slightly larger or heavier SKU and have to be picked using both hands. A type C pick is an even larger or heavier SKU that needs to be picked by bending down and using both hands to retrieve a unit. Type C SKUs would mostly be sent to the various branches as full cartons and would not necessarily enter the picking lines.

When a picker is assigned a new branch order, he/she does not have to start at the beginning of the picking line as the VRS will instruct him to the nearest SKU on his new branch order, this is known as dynamic order picking. Depending on the size of the line it takes about 1.5 to 2 days for a team of pickers to complete all the branch orders assigned to their picking line. The quantity of the left-over products are also checked after picking and compared to what should have been picked. After checking the remaining quantities the left-over stock is removed from the picking line and placed back in the storage racks.

3.5 Assigning SKUs to locations

PEP prefer that no SKUs on the picking line require replenishment during a wave of order picking as this will result in the entire picking line coming to a standstill while the SKUs are replenished. Therefore an efficient quantity of each SKU is to be placed in the picking line locations when the picking lines are constructed. In some cases the correct quantity of the SKUs for which a large volume need to be picked, does not fit in a single location on the picking line. For these instances more than one consecutive locations are allocated to the same SKU but only the first is allocated to the SKU on the system and the consecutive locations are left open due to the fact that the system is unable to recognise more than one location containing the same SKU on a picking line. The Durban DC seldom allocates more than two locations to a SKU.

The picking line manager is responsible for constructing the picking lines. This is done mostly based on his/her knowledge from experience and no mathematical optimisation is implemented during this process. A picking line consisting of 30 000 picks usually take less than a day to complete picking, while a typical picking line hardly ever contain more than 60 000 picks and would not exceed two days to complete the order picking [28].

3.6 Restrictions from PEP

Certain rules are implemented by PEP in order to ease the operations within the DC. These are mostly formed due to previous experiences and lessons learnt within the DC and may vary between DCs.

Firstly the picking lines at the PEP DCs are built according to a list of distributions (DBNs) that is received from central office. A list on the SKU level contains the number of units that each branch need to receive. A DBN may contain more than one SKU.

Each picking line has either 56 or 112 locations available to which SKUs can be assigned to and there is a limited number of picking lines available. The gate in the middle of the conveyor allows for PEP to either construct two smaller picking lines (each with 56 locations) or a single large picking line (with 112 locations). Therefore SKUs can't simply be allocated to picking lines and additional picking lines can't just be constructed in order to speed up the picking process.

In order to maintain work balance, PEP attempts to always have for each functioning picking line another picking line being built. This allows for pickers to move over to the completed picking line once all the picks on the current picking line is completed. This also assist with the work balance of the forklifts and not only the pickers.

PEP implements a FIFO stock keeping system, where the DBNs received on a specific day has to be completely picked before the next day's DBNs may be started on. The only exception to this rule is when the floor manager of the DC is informed of an urgent request requiring a specific SKU to be picked that was not scheduled. In this instance the floor manager will shift that specific DBN to be scheduled in a picking line as soon as possible.

Due to the space constraints in a picking line, PEP only allows for 8 pickers to pick the SKUs on a 56-location picking line. This prevent congestion within a picking line that could cause delays for the pickers.

One of PEPs key performance indicators (KPI) is whether or not all the SKUs contained in a DBN was dispatched within seven days of receiving the DBN. This is implemented in order to keep stock flowing through the supply chain and not sit in the DCs taking up costly storage space.

As the Durban DC also distribute to Botswana, all the SKUs designated for Botswana has to be repriced. This require that the SKUs has to go through a separate work station where the original R-value price needs to be removed by an employee and replaced with a new price tag indicating a BWP-value (pula) before it may be repackaged.

At the Durban DC products of different sizes are not allowed to be placed next to each other on the picking lines to reduce the errors during picking. This means that medium sized white boys school shirts are not allowed to be placed in the location right next to large sized white boys school shirts on a picking line.

PEP currently only allows for a SKU to be assigned to a single picking line while the VRS only allows for a SKU to be assigned to a single location on a picking line. Therefore all the branch requirements for a SKU will be picked on the same picking line. Currently if a SKU require additional space locations on a picking line, two adjacent locations is assigned to it. The VRS will recognise this as a location with stock and another location with stock. For the purpose of this study when referring to allocating multiple locations to a SKU, it is implied that these locations are non-adjacent. Therefore duplicating a SKU refer to assigning a SKU to two locations on a single picking line that are non-adjacent.

3.7 Chapter summary

This chapter gives an overview of the operations associated with and conducted within the PEPs Durban DC. A brief overview of the distribution network is described in §3.1 followed by a section on the planning process in §3.2. In §3.3 an overview of the DC layout is presented along with a description of the processes implemented in each area. An in-depth look is taken into the picking line and the order picking process in §3.4 followed by a section on the assignment of SKUs to specific locations on a picking line in §3.5. The chapter is concluded with some rules implemented by the Durban DC's management that restricts certain operations.

CHAPTER 4

Problem Description

Contents

The previous chapter discussed the operations at the Durban DC giving a better understanding of the environment on which this study is based. The planning of PEP's picking lines may be broken down into three decision tiers which ultimately aims at picking the orders in the shortest time possible. The first tier consists of considering the SKUs to be assigned to the various picking lines and then determining which SKUs are to be assigned to which picking lines. The second tier considers the positioning of the various SKUs within the picking line, while the objective of the third tier is to determine the sequence in which the orders need to be picked by the pickers. These three decision tiers where considered by De Villiers and Matthews [28, 90]. The problem definition, which is the motivation for this study, is discribed in §4.1. In §4.2 the approach to solving the SKU duplication problem is broken down into 5 questions that need to be investigated and answered.

4.1 Problem definition

Currently the WMS implemented by PEP only allows for a SKU to be placed on one picking line in one location when the DBN is released. If any of the SKUs assigned to a picking line require more space than a single location due to the volume of the SKU, 2 consecutive locations are assigned to the SKU but on the WMS it is only registered as a single location and the second location is registered as being empty. Because pickers are only allowed to walk clockwise around the conveyor belt, they are forced to pass a location at least the same number of times as the number of branches to which the SKU is to be distributed to. The number of branches to which a SKU need to be distributed to may vary anything from 1 to 1 400. Therefore if 2

SKUs are placed on the same picking line, the first SKU to be distributed to 5 branches and the second to 1 400 branches, both locations will need to be passed at least 1 400 times. Thus the first location gets passed many times without being picked from. Therefore by minimising the number of locations that are passed without picking from it will increase the pick density or picks per cycle of the picking line. One way of achieving this is by duplicating the SKUs which needs to be distributed to a large number of branches and thus decreasing the number of times a location needs to be visited to complete all the branch orders.

4.2 The SKU duplication problem

There are many ways to measure the effects of implementing new strategies in a warehouse, as may be seen from the various literature described in Chapter 2. For this study the results are captured and compared in terms of the number of cycles it will take a team of pickers to complete all the branch orders assigned to their picking line. This is sufficient as the number of cycles walked by the pickers have a positive correlation with the amount of time it will take to complete the branch orders. By decreasing the number of cycles, the time to pick the branch orders is thus decreased, resulting in less labour hours required and thus decreasing the cost of operating the picking lines. There is no need to add additional labour hours for constructing more picking lines as the same number of SKUs are removed from the storage racks and assigned to the picking lines.

SKU duplication implies that *a SKU is assigned to 2 locations on a single picking line and also registered on the WMS as occupying 2 distinct locations.* The SKU duplication problem, where a picking lines are selected and by means of duplicating SKUs increased to a total of $a + b$ picking lines (where $a \geq b$), may be divided into five subproblems, namely how many SKUs to duplicate, which SKUs to duplicate, which SKUs to remove to free up locations for the duplications, how should the new picking lines be constructed and whether to duplicate SKUs before or after assigning them to picking lines? These subproblems form the objectives for this study and are discussed in the remainder of this chapter.

4.2.1 Number of SKUs to duplicate?

The first step is to determine the distribution of the pick frequency of the SKUs for each of the original picking lines as supplied by PEP, as this will show how frequently each SKU needs to be picked to complete all the branch orders. If the majority of the SKUs have a high pick frequency, the picking line will have a high minimum number of cycles to complete all the branch orders. If only one SKU have a high pick frequency and the remaining SKUs have low pick frequencies, the minimum number of cycles to complete the branch orders will still be high. While if the SKU with the highest pick frequency is low and therefore all the other SKUs as well, the minimum number of cycles to complete the branch orders will also be low.

The significance of the distribution of the SKUs pick frequencies (when sorted from highest to lowest) is in determining where and how many "jumps" it contains. A "jump" is where there is a substantial increase in the negativity of the slope between two successive points when compared with the slope of the prior two successive points. This will give a good indication of how many SKUs need to be duplicated for each data set.

This question is twofold, firstly to determine how many of the SKU need to be duplicated and secondly how many duplications need to be on a picking line. For the first part to determine how many of the SKUs to duplicate is relevant to any of the heuristics described in Chapter 6. When duplicating all the possible SKUs one at a time and increasing the number of duplications (for a picking line with an unlimited location capacity) until all the SKUs have been duplicated, the number of cycles to complete all the branch orders reaches a plateau. In other words a point is reached where increasing the number of duplications by one does not justify the increase in the number of cycles to complete all the branch orders. To be able to determine after how many duplications this plateau occurs, will answer the question of how many SKUs to duplicate for a given group of branch orders.

The latter part of the question focus more on the number of duplications across multiple picking lines. Should they be split equally or will there be a greater saving in the number of cycles to complete the branch orders if there is an unequal number of duplicated SKUs on each of the picking lines.

4.2.2 Which SKUs to duplicate?

Parallel to the problem of determining the number of SKUs to duplicate in a picking line, is to determine which is the specific SKUs that must be duplicated. These two aspects cannot be viewed in isolation as the properties and characteristics of the specific SKUs will determine the number of SKUs to duplicate. The "jumps" will occur most likely at the point where duplicating one additional SKU will not justify the increase in saving in the number of cycles to complete all the branch orders.

In addition, the possibility may arise that the optimal way to duplicate products in all the original picking lines is by splitting the SKUs so that half go to the first picking line and the second half to the next picking line and filling each of the picking line by duplicating all the SKUs.

4.2.3 Which SKUs to remove from the picking line to free up locations?

When SKUs are duplicated on picking lines, the number of locations needed increase. As PEP's picking lines have a maximum capacity of 56 locations to which SKUs may be assigned on each picking line, it will require that SKUs be removed so that the maximum capacity is not exceeded. Another problem thus arises. When duplicating a certain number of SKUs on a picking line, which SKUs will need to be removed from the picking line to be able to accommodate all the duplications. Once established how many SKUs to duplicate and which SKUs to duplicate on a picking line, as well as how many and which SKUs to remove from the picking line to create space for the duplications, the next problem that arise is where (i.e. in which picking line) the removed SKUs need to be picked.

4.2.4 How to assemble the new picking lines?

There are many ways to construct the new picking lines. This problem arise when additional picking lines is created from the original picking lines to accommodate the increase in the number of locations required due to SKUs being duplicated. The problem therefore is how to go about assigning SKUs and their duplications to the multiple picking lines. Implied in this problem is which SKUs need to go together on a picking line.

To answer this question, various heuristics are formulated to construct picking lines in various ways, taking into account the duplications and capacity of the picking lines. The results are compared in terms of which heuristic yields the greatest saving in the number of cycles it would take the pickers to complete all the branch orders.

4.2.5 Duplicate SKUs before or after assigning them to picking lines?

It is also necessary to determine whether SKUs should be duplicated before assigning them to a picking line and then duplicating a more or less equal number of SKUs on each picking line or whether to duplicate SKUs first and only then assigning them to the various number of picking lines. This will indicate whether assigning an equal or unequal number of duplicated SKUs to each picking line yields better results with regards to the travel distance of the pickers.

4.3 Chapter summary

As discussed in this chapter the questions that need to be answered in this study is:

1. What is the number of SKUs to duplicate?
2. Which SKUs must be duplicated?
3. Which SKUs must be removed from the line?
4. How should the new picking lines be assembled?
5. Should SKUs be duplicated before or after assigning them to picking lines?

There are also the question of where to place duplicated SKUs on the picking lines. Scott [119] has concluded that the actual location that a SKU is assigned to does not have a very significant influence on the minimum number of cycles it would require to complete all the branch orders for the relevant picking line. Therefore the SKU to location allocation problem, where the exact location that each SKU is to be assigned to on a picking line is to be determined, is not considered for this study.

CHAPTER 5

Experimental exploration of SKU duplication

Contents

Utilising the knowledge of PEP, their process flow in the Durban DC and considering the problem to be solved, the next step is to analyse the original picking lines supplied by PEP in order to achieve a better understanding of the characteristics of these original picking lines. In this chapter the effect of duplicating specific and all SKUs on the original picking lines is experimented with. This is achieved by considering scenarios where the picking line has an unlimited as well as a limited capacity.

5.1 Original picking lines

PEP supplied twenty-two original picking lines they constructed to pick SKUs for distribution to their branches across South Africa. These original picking lines varied in the number of SKUs, the number of branches receiving the products and also the distribution of the pick frequency of each SKU assigned to the different picking lines.

Based on the number of locations used to construct the picking line and the highest pick frequency of the SKUs, these picking lines received from PEP were categorised as either small, medium or large picking lines, as shown in Table 5.1. Of the 22 original picking lines, 5 were categorised as small (original picking lines R, S, T, U and V), 7 as medium (original picking lines K, L, M, N, O, P and Q) and 10 as large (original picking lines A, B, C, D, E, F, G, H, I and J). The five original picking lines categorised as small are too small to analyse and deduce valuable insights and are thus ignored for the most part of this study.

Size category	Original picking line	# SKUs	Highest pick frequency	Minimum cycles	Branches serviced
Large	A	49	1232	1232	1262
	B	54	1226	1226	1264
	C	51	1161	1161	1265
	D	56	1011	1022	1263
	E	51	1069	1070	1264
	F	55	896	882	1258
	G	53	959	1007	1258
	H	54	977	966	1244
	I	56	855	951	1260
	J	56	729	941	1264
Medium	K	63	95	230	943
	L	56	141	231	846
	M	51	150	151	728
	N	55	117	115	733
	O	63	74	91	396
	P	48	67	81	574
	Q	64	33	45	242
Small	R	55	13	15	158
	S	42	9	9	89
	T	51	8	8	82
	U	48	7	7	90
	V	56	5	5	80

Table 5.1: *Size category for the original picking lines A to V based on number of SKUs and highest pick frequency of SKUs, and minimum cycles to pick branch orders.*

The current method of assigning SKUs to picking lines implemented by PEP, is that they first sort the SKUs from highest pick quantity to lowest after receiving the DBNs from head office. Therefore the SKU in the batch that need to be assigned to a picking line, with the highest total pick quantity are assigned to the first picking line, the SKU with the next highest number of picks, gets assigned to the next picking line, and so forth until all SKUs that has a DBN are assigned to a picking line. Only after the SKUs have been assigned to the various picking lines is the branch orders constructed based on the SKUs and their DBNs for the respective picking lines. This is done so that the total number of picks on each picking line is more or less equal, giving the impression that the work is distributed evenly across the picking lines and thus each picker will have an even workload. The total number of picks are counted and not the pick frequency, thus if 5 units of a SKU needs to be distributed to a branch, it is counted as 5 picks and in the case of pick frequency, it is counted as 1 branch.

Each picking line has a capacity of 56 locations, thus a batch of branch orders consisting of 56 SKUs may be assigned to a picking line for picking. From Table 5.1 it is clear that branch orders are not always batched in such a way that SKUs will fill all the locations, leaving some locations open and unused during picking. In some cases these empty locations on the picking line did contain SKUs as the SKU required a second location so that the correct quantity would fit onto the picking line, but because of the WMS limitations these locations are captured as empty. As for the purpose of this study these locations are considered to be empty. From the 22 original picking lines only 5 have 56 SKUs assigned to the picking line (original picking lines D, I, J, L and V) and 3 have more than 56 SKUs assigned to them (original picking lines K, O and Q). For the original picking lines containing more than 56 SKUs, PEP would pick the excessive SKUs first which is only destined for a few branches and once these SKUs are completely picked they continue in the normal fashion with the remaining 56 SKUs. These excessive SKUs are usually PEP's home products which only goes to the their home stores of which there is very few branches in South Africa and can therefore be picked quickly. The other 14 original picking lines all contain less than 56 SKUs and therefore have empty locations when branch orders are being picked. These empty locations are usually used to store empty cartons which the pickers use to pack the branch orders in.

The pick frequency of a SKU refers to the number of branches to which a SKU needs to be distributed to. This is also the number of times a location needs to be visited to complete all the branch orders assigned to the picking line. The SKU with the highest pick frequency also gives a fair lower bound to the number of cycles that needs to be walked by the pickers to complete all the branch orders as the pickers have to pass that location at least the frequency number of times. For instance the highest pick frequency for all SKUs in original picking line A is 1232 times. Thus the location containing this SKU needs to be visited 1232 times and the pickers will have to complete at least 1232 cycles to be able to walk past that location 1232 times as they are only allowed to walk clockwise around the conveyor belt. The minimum number of cycles to complete the branch orders may be greater than or equal to the highest pick frequency, depending on the correlation between branches requiring the different SKUs.

When running the original picking lines without any adjustments through the Picking Line Solver developed by De Villiers and Matthews [28,90] it gives the minimum number of cycles to complete all the branch orders assigned to the picking line as indicated in Table 5.1. From the results in Table 5.1 it is clear that for almost all of the 22 original picking lines, the minimum number of cycles to complete all the branch orders is either equal to or greater than the highest pick frequency of the SKUs for the respective original picking lines. The number of branches is always greater than the number of cycles and highest pick frequency for all the original picking lines, meaning none of the SKUs are distributed to all the branches for the respective original picking lines.

Figure 5.1 contains the pick frequencies of the SKUs (sorted from highest to lowest pick frequency) for all the large original picking lines. Figure 5.2 contains the same pick frequencies of the SKUs for the medium original picking lines. There is no uniformity between the distribution of the various SKUs for the different original picking lines. The pick frequency of some of the original picking lines decrease quickly – for example original picking line B. Others decrease gradually – for example original picking line H, or have a slow decrease in the pick frequency of the SKUs – for example original picking line A. It is clear that all the original picking lines start decreasing at different points and at a different rate, thus no two original picking lines have the exact same SKU pick frequency graph, but all display fairly similar trends.

5.2 Duplicating SKUs on a picking line

Currently PEP's systems are only able to handle placing a SKU on a picking line once. By duplicating a SKU so that it occupy two locations on a picking line instead of one, the pick frequency of the duplicated SKU may be halved. Thus it can change the distribution of the SKUs pick frequency on the picking line.

The SKU with the highest pick frequency forms the lower bound on the number of cycles

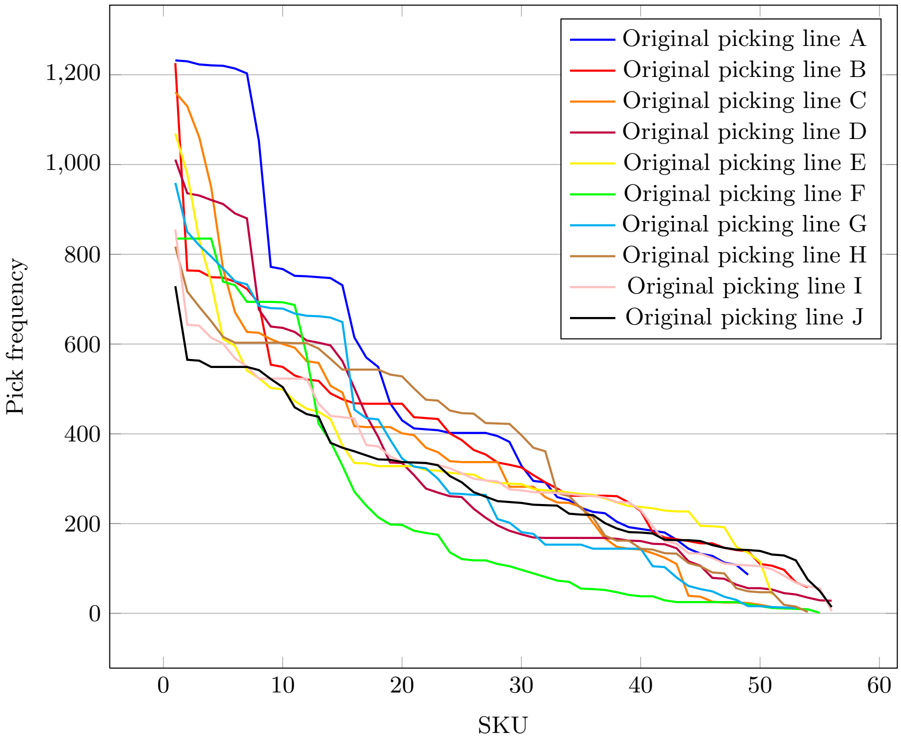


Figure 5.1: *The pick frequency of SKUs for all the large original picking lines.*

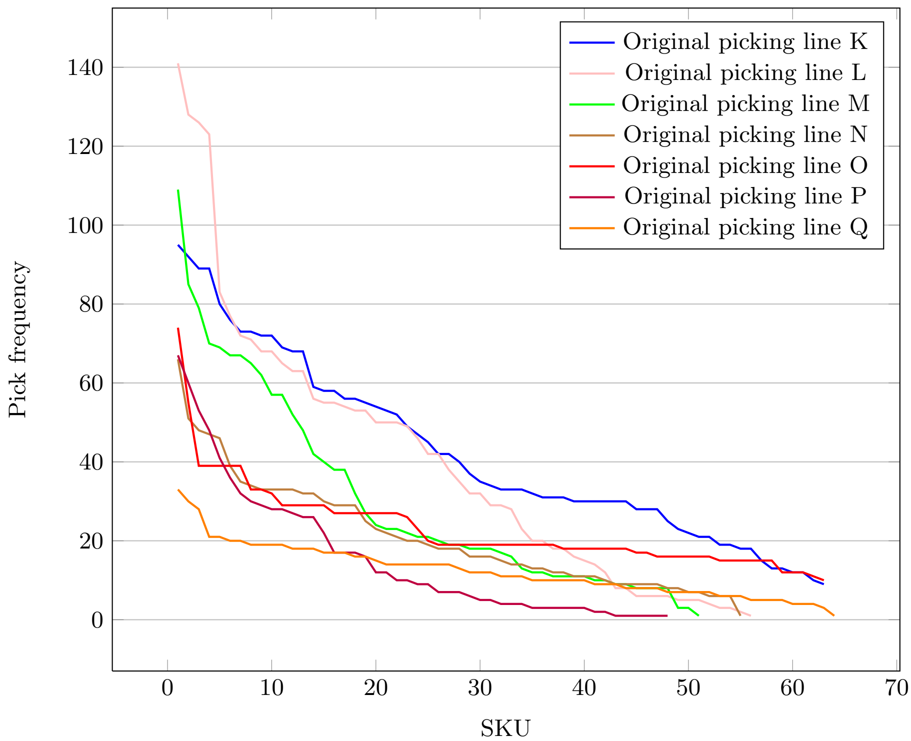


Figure 5.2: *The pick frequency of SKUs for all the medium original picking lines.*

required to complete all the branch orders. Thus by duplicating the SKU with the highest pick frequency, the highest pick frequency may be halved which influence the minimum number of cycles it would take to complete all the branch orders. If starting by duplicating another SKU which does not have the highest pick frequency, the minimum cycles would still be equal to or greater than the highest pick frequency. Therefore a good strategy may be to start by duplicating the SKU with the highest pick frequency first, and in the case where there is more than one duplication the SKU with the next highest pick frequency is duplicated and then the next highest and so forth.

5.2.1 Duplicating each SKU individually

Duplicating each SKU individually implies that 54 picking lines are constructed from the original picking line B (with 54 SKUs) each containing one distinct duplicated SKU. On each picking line a different SKU is duplicated. Solving these picking lines by means of the Picking Line Solver as developed by De Villiers and Matthews [28, 90], yields a result where the picking line on which the SKU with the highest pick frequency is duplicated can be picked with 972 cycles. All the other picking lines can be picked by the pickers in 1226 cycles. Thus the number of cycles is only decreased when the SKU with the highest pick frequency is duplicated and no other SKU. On the original picking line all the branch orders could be completed by the pickers in 1226 cycles. This is not proof, but supports the idea that it is best to duplicate the SKUs with the highest pick frequency first.

Table 5.2 shows the minimum and maximum number of cycles required when duplicating one SKU at a time for each of the possible SKUs for all the large and medium original picking lines. It also compare the minimum and maximum number of cycles with the number of cycles required when duplicating SKU 1, the SKU with the highest pick frequency. In total, of the 10 large original picking lines provided with, when only duplicating one SKU and not removing any SKUs, 6 of the original picking lines (original picking lines B, E, G, H, I and J) yielded the minimum number of cycles when SKU 1 was duplicated (giving SKU 1 a ranking of 1). For the original picking lines A, C and F duplicating SKU 1 gave the second lowest number of cycles to complete all the branch orders (therefore SKU 1 is ranked second). Duplicating SKU 1 only for the original picking line D, was outperformed by 15 other SKUs.

Of the 7 original picking lines categorised as medium, only for the original picking lines K, N and O did duplicating SKU 1 yield the minimum number of cycles (ranking SKU 1 first). For original picking line P when duplicating SKU 1 gave the second lowest number of cycles (SKU 1 is ranked second), while for original picking line M it yielded only the fourth lowest number of cycles to complete the branch orders (therefore SKU 1 is ranked fourth). For the medium original picking lines L and Q, duplicating SKU 1 was outperformed by 36 and 39 other SKUs respectively.

5.2.2 Filling non-full picking lines by duplicating SKUs

The original picking lines in Table 5.3 was originally not filled to capacity, meaning they do not contain 56 SKUs. This creates a situation where there is empty locations on a picking line that the pickers pass, as PEP's picking lines have a fixed capacity of 56 locations each. The number of SKUs on these picking lines vary from 48 to 55, and the picking lines need additional SKUs

Original picking line	Minimum number of cycles	SKU	Maximum number of cycles	SKU 1 cycles	SKU 1 rank
A	1230	11	1233	1232	2
B	972	1	1226	972	1
C	1130	3	1165	1161	2
D	973	2	1036	1018	16
E	1004	1	1079	1004	1
F	854	12	981	872	2
G	968	1	1014	968	1
H	955	1	978	955	1
I	924	1	958	924	1
J	921	1	943	921	1
K	223	1	243	223	1
L	219	3, 27	233	230	36
M	137	6	156	144	4
N	106	1	115	106	1
O	75	1	95	75	1
P	75	8	87	76	2
Q	38	5, 24, 43, 56	49	43	39

Table 5.2: *Minimum and maximum number of cycles required to pick all the branch orders for all the large (A – J) and medium (K – Q) original picking lines, when duplicating one SKU without removing any from the original picking lines, compared with the number of cycles when duplicating SKU 1 (it is the SKU with the highest pick frequency) for each of the original picking lines.*

to fill all the locations or some of the SKUs already assigned to the picking line can be assigned to a second location on the same line.

When duplicating the SKUs with the highest pick frequencies so that all the locations on the picking line get a SKU assigned to it, creates a saving in the number of cycles the pickers has to complete, as indicated in the last column of Table 5.3. The savings achieved by duplicating SKUs with the highest pick frequencies on a picking line, vary from 0,56% to 32,84%. The more SKUs that are duplicated on the picking line, the greater the saving in the number of cycles to be completed by the pickers. The saving in the number of cycles is influenced by the correlation between branch orders and the SKUs, the distribution of the pick frequency of the SKUs.

In Figure 5.1 it is visible that there is a substantial increase in the slope between the SKUs with the highest and eighth highest pick frequency, which supports the 13,72% saving when duplicating the seven SKUs with the highest pick frequencies in the original picking line A. In Figure 5.1 it can also be seen that there is a minor change in the slope between the SKU with the highest and second highest pick frequencies and therefore a saving of only 0,56% in the number of cycles is incurred when duplicating only the SKU with the highest pick frequency in the original picking line F. For the original picking line P, (Figure 5.2) there is a minimal change in the gradient between the SKUs with the eighth and ninth highest pick frequencies, but a substantial change in the slope between the SKU with the highest and ninth highest pick frequencies. Thus the conclusion may be made that the slope between the SKU with the highest pick frequency and the first SKU to not be duplicated, has to be substantial to have a noteworthy saving in the total number of cycles.

Original picking line	Nr SKUs	Original cycles	Nr duplications	Cycles with duplications	% Saving
A	49	1232	7	1063	13.72
B	54	1226	2	991	19.17
C	51	1161	5	945	18.60
E	51	1069	5	929	13.10
F	55	896	1	891	0.56
G	53	959	3	936	2.40
H	54	977	2	955	2.25
M	51	150	5	116	22.67
N	55	117	1	105	10.26
P	48	67	8	45	32.84

Table 5.3: *Percentage saving in the number of cycles to complete the branch orders for all the large (original picking lines A, B, C, E, F, G and H) and medium (original picking lines M, N and P) original picking lines not filled to capacity.*

5.3 Multiple duplications on a picking line

When duplicating SKUs on picking lines there are one of two possible ways to look at it, either considering the case where the picking lines have unlimited capacity or when there is a limited capacity for the picking lines as is the case at PEP.

5.3.1 Picking lines with unlimited capacity

For picking lines with unlimited capacity, all SKUs can be duplicated and none has to be removed from the picking line to create space for the duplications. This create a pick frequency distribution where all the SKUs pick frequencies is half of the original and all the SKUs are allocated to two locations on the picking line.

Table 5.4 shows a comparison of the number of cycles to complete the branch orders when duplicating none of the SKUs (0 duplications) and when duplicating all of the SKUs in the original picking line (max duplications). The last two columns shows the number of SKUs that need to be duplicated to achieve the minimum number of cycles to complete the branch orders and what the minimum number of cycles is if there is an unlimited number of locations available on the picking lines. The SKUs with the highest pick frequency is always duplicated first, thus when duplicating 10 SKUs means the 10 SKUs with the highest pick frequencies are duplicated.

This means that for the original picking line A, which contains 49 SKUs, the original number of cycles to complete all the branch orders is 1232. When assigning all the SKUs to two locations the minimum number of cycles to complete the branch orders is 617, which yields a 50,1% saving in the number of cycles with a 100% increase in the locations required and therefore the distance that the pickers need to walk per cycle. This is also the minimum number of cycles over all the possible number of duplications. In the case of original picking line I, containing 56 SKUs, the original number of cycles to pick all the branch orders is 953, while when duplicating all the SKUs require completing 522 cycles by the pickers. However, the minimum number of cycles to complete the branch orders is achieved by duplicating the 50 SKUs with the highest pick frequency.

Figures 5.3 (the large picking lines) and 5.4 (medium picking lines) shows the decrease or in-

Original picking line	Nr of SKUs	Number of cycles 0 Duplications	Max Duplications	Minimum	SKUs duplicated at minimum
A	49	1232	617	617	49
B	54	1226	616	613	51
C	51	1163	584	584	50
D	56	1018	523	523	56
E	51	1070	545	545	51
F	55	896	450	445	54
G	53	999	554	526	44
H	54	977	542	531	51
I	56	953	522	520	50
J	56	935	476	476	56
K	63	243	107	104	62
L	56	218	104	101	51
M	51	150	68	65	50
N	55	117	48	45	46
O	63	82	45	39	62
P	48	80	38	34	38
Q	64	48	23	19	57

Table 5.4: *Minimum number of cycles when duplicating none to all SKUs for the medium (original picking lines K – Q) and large (original picking lines A – J) original picking lines.*

crease in the number of cycles to complete the branch orders, when duplicating an additional SKU on a picking line with unlimited capacity. There is an overall decreasing trend with an increase in the number of SKUs to duplicate, but at times when an additional SKU is duplicated there is an increase in the number of cycles for the pickers. The maximum saving is also less for the original picking lines requiring fewer cycles to complete the branch orders, than for the original picking lines for which the branch orders can be completed with more cycles.

By duplicating SKUs the number of locations assigned to the same number of SKUs increase. As no SKUs are being removed from the picking line when duplicating, the number of locations on the picking line increase, in other words the length of the picking line increases and thus the distance the picker has to walk for each cycle and therefore the time it takes to complete a cycle increases. This creates a trade-off between decreasing the number of cycles and increasing the locations on the picking line. To see the actual effect of duplicating SKUs on a picking line on the number of cycles the picker has to complete, a picking line with limited capacity need to be investigated where excess SKUs can be assigned to the next picking line.

5.3.2 Picking lines with limited capacity

PEP's picking lines have a limited capacity of 56 locations, therefore restricting the number of duplications possible for each of the original picking lines. Table 5.5 shows the minimum number of cycles to complete all the branch orders for the original picking lines and when duplicating the SKUs with the highest pick frequency and removing the same number of SKUs as duplicated. For this case the SKUs with the highest pick frequency was duplicated but all possible SKUs was removed to determine which SKUs need to be removed to give the minimum number of cycles for each of the original picking lines. Tables 5.6 and 5.7 show which SKUs had to be removed to minimise the number of cycles to pick all the branch orders for all the original picking lines when duplicating one and two SKUs respectively.

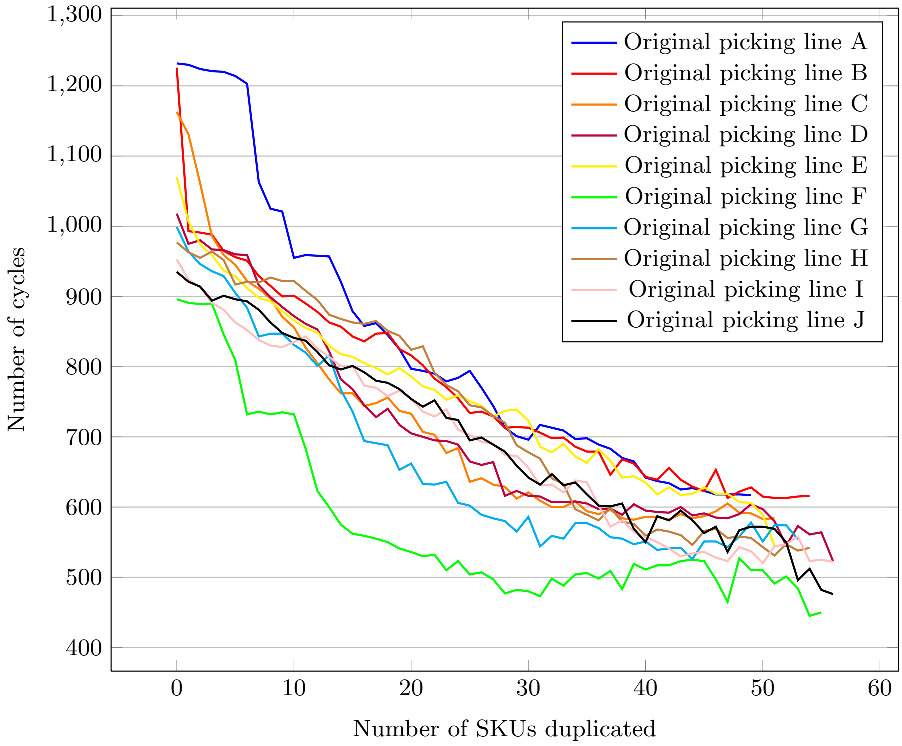


Figure 5.3: *Number of cycles when duplicating multiple SKUs for all the large original picking lines, without removing any SKUs from picking line with unlimited capacity.*

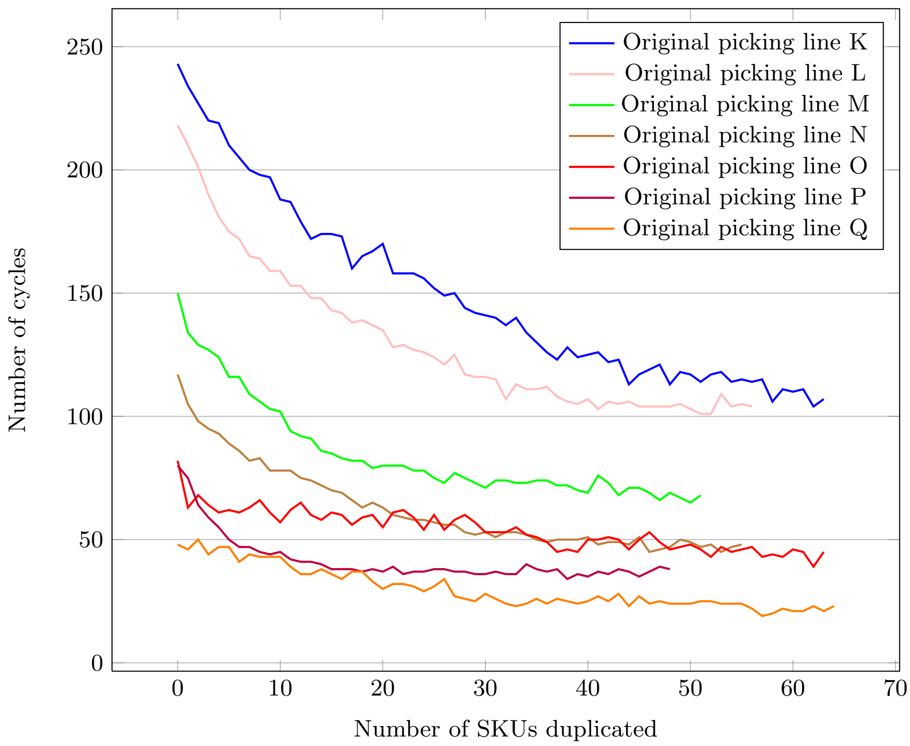


Figure 5.4: *Number of cycles when duplicating multiple SKUs for all medium original picking lines, without removing any SKUs from picking line with unlimited capacity.*

When duplicating one SKU and removing one, there is a saving on all original picking lines between 0% and 44.44% on the original number of cycles required to complete the branch orders, with an average saving of 14.71%. For the original picking lines, excluding the ones classified as small, there is a saving of between 0.73% and 32.97%, with an average saving of 11.02%. There is an even bigger saving of between 0.89% and 44.44% on all the original picking lines when duplicating and removing two SKUs, with an average saving of 20.22%. With a saving of between 0.89% and 38.27% for the original picking lines excluding the 5 original picking lines categorised as small, which yields an average saving of 16.41% on the number of cycles.

Tables 5.6 and 5.7 shows which SKUs was removed (*SKU removed* columns) from the original picking lines when duplicating the same number of SKUs, to give the minimum number of cycles to complete the branch orders for the respective original picking lines. There is also a column showing the SKU rank. The difference between these two columns is that the SKU removed is the position of that SKU on the list whilst the SKU rank is the actual ranking and thus more than one SKU can have the same ranking if they have the same pick frequency.

For the 17 medium and large original picking lines, when one SKU is duplicated and removed, the SKU with the next highest pick frequency is removed in 9 cases, see Table 5.6. For a further 2 original picking lines the SKU removed was under the top 5 ranked SKUs and for the remaining 6 original picking lines, the SKU that was removed was ranked 6 or higher. When duplicating 2 SKUs and thus also removing 2, for 7 of the 17 medium and large original picking lines the 2 SKUs with the next highest pick frequencies was removed. While for another 3 original picking lines both the SKUs that was removed, was ranked under the top 5 highest pick frequencies. A further 5 original picking lines had one of the removed SKUs under the top 5 and one ranked sixth or higher, while the remaining 2 original picking lines had both SKUs that was removed ranked sixth or higher.

In Table 5.7 the two SKUs is displayed that have to be removed for the number of cycles to be minimised when the 2 SKUs with the highest pick frequency is duplicated. For the original picking lines A, R, S, T and U there is 26, 84, 653, 976 and 45 possible combinations of two SKUs that can be removed to achieve the minimum number of cycles. The original picking lines R, S, T and U is categorised as small (see Table 5.1), therefore it is expected that there will be a high number of possible solutions. As for the original picking line A, the SKUs to remove is 26 combinations which all include SKU 3. For the original picking lines A, R, T and U, removing SKU 3 and 4 was included in all the possible solutions and for the original picking line S removing SKUs 3 and 5 was included in all the possible solutions.

As the number of SKUs to duplicate increase and thus also the number of SKUs to remove to maintain the same number of locations on the line as the original picking lines, the number of possible combinations of SKUs also increase. Table 5.8 shows the possible number of combinations when duplicating and removing one and two SKUs. It is clear the possible combinations increase drastically with the increase in number of SKUs to duplicate and remove, and therefore also the time to solve all the possible combinations. Thus it is too time consuming to solve all possible combinations for each original picking line to learn how many SKUs need to be duplicated and which SKUs to remove to minimise the number of cycles to pick each of the original picking lines. Furthermore the SKUs that are removed need to be added to another picking line so that they may be picked for distribution as well.

Original	Minimum cycles			% Saving	
picking line	0 SKUs	1 SKU	2 SKUs	1 SKU	2 SKUs
A	1232	1223	1221	0.73	0.89
B	1226	974	968	20.55	21.04
C	1161	1063	947	8.44	18.43
D	1022	962	942	5.87	7.83
E	1070	960	933	10.28	12.80
F	882	855	834	3.06	5.44
G	1007	955	903	5.16	10.33
H	966	948	917	1.86	5.07
I	951	912	885	4.10	6.94
J	941	904	880	3.93	6.48
K	230	221	207	3.91	10.00
L	231	198	172	14.29	25.54
M	151	124	112	17.88	25.83
N	115	95	87	17.39	24.35
O	91	61	57	32.97	37.36
P	81	62	50	23.46	38.27
Q	45	43	37	4.44	17.78
R	15	9	9	40.00	40.00
S	9	5	5	44.44	44.44
T	8	5	5	37.50	37.50
U	7	6	5	14.29	28.57
V	5	5	4	0.00	20.00

Table 5.5: *Minimum number of cycles to pick the original picking lines when duplicating and removing 0, 1 and 2 SKUs from original picking lines. The last two columns show the percentage saving in number of cycles on the original picking lines. The large picking lines are original picking lines A – J, the medium picking lines are original picking lines K – Q, while the original picking lines R – V are the small picking lines.*

Original picking line	SKU removed	SKU ranking	Pick frequency	Min cycles
A	2	2	1230	1223
B	16	16	468	974
C	2	2	1130	1063
D	10	10	636	962
	34	32	168	962
E	2	2	979	960
F	6	3	731	855
G	12	12	663	955
H	5	5	616	948
I	2	2	643	912
J	2	2	565	904
K	11	8	69	221
L	2	2	128	198
	4	4	123	198
M	2	2	85	124
N	2	2	51	95
O	45	13	17	61
P	2	2	60	62
Q	8	6	19	43
	31	13	12	43
	60	21	4	43
R	2	2	10	9
S	10	3	4	5
T	2	2	6	5
U	2	1	7	6
V	2	1	5	5
	4	1	5	5
	5	2	4	5
	7	2	4	5
	33	4	2	5
	34	4	2	5
	40	5	1	5
	47	5	1	5

Table 5.6: *SKU that need to be removed to obtain minimum cycles when duplicating the SKU with the highest pick frequency and removing another SKU. The large picking lines are original picking lines A – J, the medium picking lines are original picking lines K – Q, while the original picking lines R – V are the small picking lines.*

Original picking lines	SKUs removed	SKUs ranking	Pick frequency	Min cycles
A	×26			1221
B	7, 16	7, 16	723, 468	968
C	3, 4	3, 4	1061, 950	947
D	13, 45	13, 40	603, 106	942
E	3, 4	3, 4	832, 738	933
F	3, 4	1, 1	835, 835	834
G	3, 4	3, 4	820, 795	968
H	5, 13	5, 8	616, 590	917
I	3, 5	3, 5	641, 601	885
J	8, 21	5, 18	542, 336	880
K	6, 26	5, 20	76, 42	207
	6, 41	5, 28	76, 30	207
	14, 21	10, 15	59, 53	207
L	3, 4	3, 4	126, 123	172
M	3, 5	3, 5	79, 69	112
N	3, 4	3, 4	48, 47	87
	3, 5	3, 5	48, 46	87
O	10, 31	5, 11	32, 19	57
	38, 45	12, 13	18, 17	57
P	3, 5	3, 5	53, 41	50
Q	3, 27	3, 11	28, 14	37
	7, 48	5, 18	20, 7	37
	11, 48	6, 18	19, 7	37
	27, 28	11, 12	14, 13	37
R	×84			9
S	×653			5
T	×976			5
U	×45			5
V	3, 4	1, 1	5, 5	4

Table 5.7: *SKUs that need to be removed to obtain minimum cycles when duplicating the two SKUs with the highest pick frequency and removing another two SKUs. The large picking lines are original picking lines A – J, the medium picking lines are original picking lines K – Q, while the original picking lines R – V are the small picking lines.*

Original picking lines	1 SKU	2 SKUs
A	48	1081
B	53	1326
C	50	1176
D	55	1431
E	50	1176
F	54	1378
G	52	1275
H	53	1326
I	55	1431
J	55	1431
K	62	1830
L	55	1431
M	50	1176
N	54	1378
O	62	1830
P	47	1035
Q	63	1891
R	54	1378
S	41	780
T	50	1176
U	47	1035
V	55	1431

Table 5.8: *The number of possible combinations of SKUs to remove when duplicating the same number of SKUs with the highest pick frequency. The SKUs that are removed is not brought into account. The large picking lines are original picking lines A – J, the medium picking lines are original picking lines K – Q, while the original picking lines R – V are the small picking lines.*

5.4 Assumptions

It is necessary to make certain assumptions for the purpose of this study due to processes imposed by PEP and constraints as discussed in §3.6.

1. There is a fixed number of picking lines and locations on each picking line in PEPs Durban DC. There are 5 picking lines (consisting of either two 56-location picking lines or a single 112-location picking line) and 1 permanent picking line for the non-seasonal items. Therefore no consideration is given to a good mix of picking lines and locations on each. For the purpose of this study a picking line refer to the 56-location picking line and not to a 112-location picking line or the permanent picking line.

2. It is assumed that the time it takes to build a picking line is significantly less than the time it takes to complete all the picks on a picking line. This ensures that the order picking process is not held up by the building of a picking line and pickers do not have idle time while working.

3. A SKU may be assigned to a max of 2 locations (never more than 2).

4. When duplicating there should always be SKUs assigned to all the locations and none should be left empty/unused. Thus the minimum number of additional picking lines to construct from the selected original picking lines is 1 and the maximum is equal to the number of original picking lines selected, as each SKU cannot get assigned to more than 2 locations. For instances where the original picking lines selected does not contain 56

SKUs originally, the maximum number of additional picking lines to construct is equal to 1 less than the number of original picking lines selected, because duplicating all the SKUs will not fill all the locations.

5. The required quantity for all SKUs are able to fit into a single location on a picking line. It is not neccessary to assign two locations to a SKU for it to fit without requiring replenishments during the picking wave.
6. The number of pickers assigned to a picking line is determined by management based on experience and is usually 8 pickers per picking line. The problem to determine the number of pickers per picking line therefore falls outside the scope of this study.
7. Pickers always walk clockwise around the conveyor belt only. This prevent congestion and therefore eases the picking process and is enforced by the VRS.
8. The demand for each SKU is predetermined and does not change while the picking line is in operation.

5.5 Chapter summary

From the experimental duplications as discussed in this chapter, it is clear that there are certain savings possible in terms of travel distance of the pickers to complete the order picking by duplicating certain SKUs on a picking line. From the results discussed in §5.2.1 it is clear that the SKUs with the highest pick frequency should be duplicated first as the pick frequency of the SKUs forms a lower bound on the number of cycles it would take the pickers to complete the order picking. The results discussed in §5.2.2 illustrate that by assigning duplicate SKUs to the unused locations even more savings are possible in terms of the travel distance of the pickers.

The results in §5.3.1 illustrate that with an increase in the number of locations required by the SKUs and therefore also an increase in the distance of each cycle, there is a decrease in the number of cycles that the pickers have to complete to make all the picks. Therefore there is a trade-off between the number of SKUs to duplicate and the travel distance per cycle for the pickers. When 2 SKUs are duplicated (the 2 with the highest pick frequency on the picking line), 2 SKUs are required to be removed from the picking line in order to maintain an equal number of locations required by the SKUs on the picking line. From §5.3.2 the conclusion can be made that removing the 2 SKUs with the next highest pick frequency bring about the greatest saving in the number of cycles that the pickers will have to complete. It was also concluded that it would be too time consuming to try every possible combination of duplicating and removing SKUs in order to determine the combination of duplications and removing SKUs that would yield the minimum number of cycles. Thus algorithms is suggested in order to decrease the number of cycles by duplicating and assigning SKUs to picking lines.

CHAPTER 6

Algorithms

Contents

With a better understanding of the PEP environment, the problem at hand as well as some characteristics of the data, various duplication approaches can now be discussed which will assist in achieving the objective of improving the number of cycles walked by the pickers to complete all the branch orders assigned to the picking line. By minimising the number of cycles walked by the pickers, the branch orders are picked faster and more effectively saving on the operating cost of the DC.

As the data sets received from PEP consists of already constructed picking lines, the option is to either use these picking lines as a starting point or to undo the work already done to construct the picking lines by combining all the DBNs on the various data sets into a single DBN list and assigning the SKUs to picking lines irrespective of the picking lines that they were previously assigned to by PEP. Therefore in §6.2 Algorithm 1 and 2 are discussed which use the data sets received from PEP as a starting point. Algorithm 1 first duplicate the SKUs with the highest pick frequency and then remove the non-duplicated SKUs with the next highest pick frequency. Algorithm 2 on the other hand duplicate the SKUs with the highest pick frequencies and then remove the SKUs (duplicated and non-duplicated) with the highest pick frequency per location.

The remainder of the algorithms discussed, first combine all the DBNs from the various selected data sets into a single DBN list and assign the SKUs to picking lines irrespective of the previous picking lines. In §6.3 the SKUs with the highest pick frequencies are duplicated, then the picking lines are constructed by assigning the SKUs to the picking lines based on their pick frequency per location. Algorithm 3 assign the duplicated and non-duplicated SKUs separately, whereas Algorithms 4, 5 and 6 assign them simultaneously. The order in which the SKUs are assigned to the picking lines differ between the algorithms. Algorithms 3 and 4 assign the SKUs by cycling over the picking lines while Algorithm 5 group the SKUs with the highest pick frequency per bin location together and then the SKUs with the next highest. Algorithm 6 sequentially assign the SKUs with the highest and lowest pick frequency per bin location to the first picking line and the SKUs with the next highest and lowest pick frequency per bin location to the next picking line.

The algorithms in §6.4 serves to investigate whether it is more efficient to duplicate the SKUs with the highest pick frequencies on the combined DBN list and assigning them to the picking lines based on their pick frequency per location, or to assign them to the picking lines first based on their pick frequency and then duplicating the SKUs on the individual picking lines with the highest pick frequency. Thus for Algorithms 7–9 the SKUs are assigned to the various picking lines first based on their pick frequency and only then a specified number of SKUs on each picking line with the highest pick frequency are duplicated. However, they differ in the method of how SKUs are assigned to the picking lines.

The algorithms discussed in §6.5 is two fold in the sense that a seed branch or SKU is identified and assigned to the first picking line. Then the additional SKUs to be added to the seed is determined based on the clustering rule for the specific algorithm, and assigned to the same picking line to enlarge the cluster containing the seed. Algorithm 10 and 11 in §6.5.1 and 6.5.2 respectively, cluster the SKUs based on branches. The rule to determine the seed branches are the same but the clustering measure differ. Algorithm 12 and 14 use a similarity matrix to assign the SKUs with the maximum mutual branches to the picking lines first, while Algorithm 13 and 15 implement a dissimilarity matrix to assign the SKUs with the minimum mutual branch variance to the picking lines first. Algorithm 12 and 13 first fill the first picking line before moving on to the next while Algorithm 14 and 15 visit the picking lines in a cyclical fashion when assigning SKUs.

6.1 Decision flow of the algorithms

There are 4 questions that need to be answered when constructing the algorithms presented here. The answers will determine how the specific algorithm handles the selected data (original picking lines). Figure 6.1 gives a schematical representation of the decision flow when constructing the algorithms. From this it is clear that for some of the algorithms 4 decisions is to be made

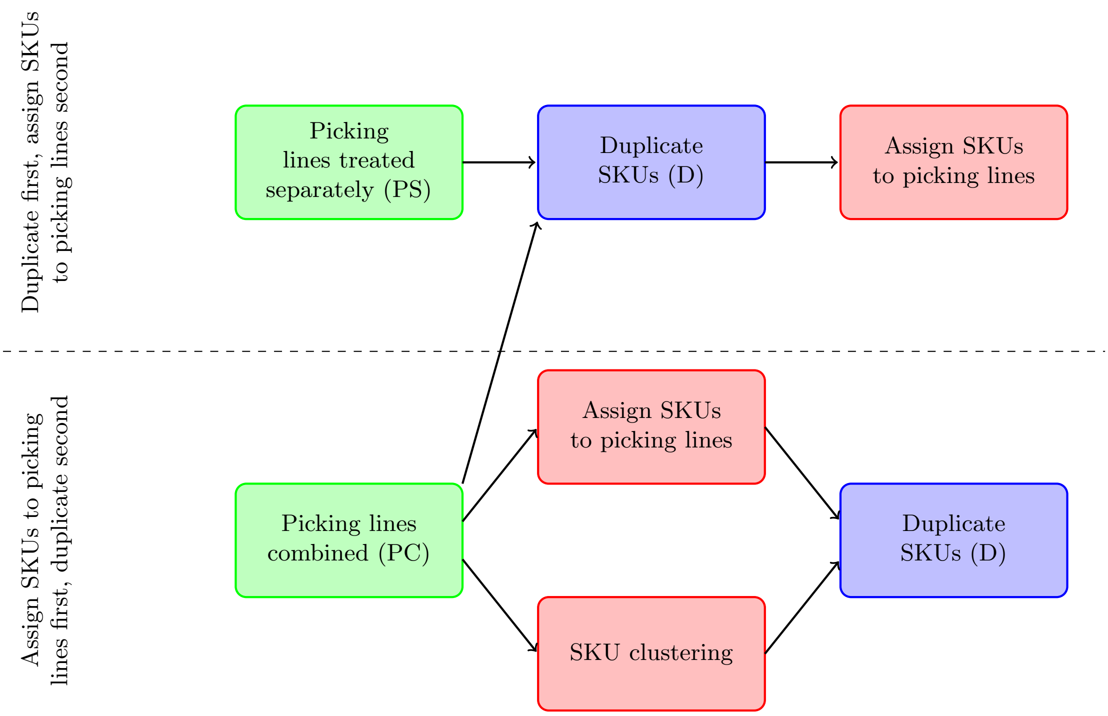


Figure 6.1: *A schematical representation of the decision flow when constructing the algorithms. Question 1 determines whether the original picking lines are to be treated separately (PS) or combined (PC). The second question is to decide if the SKUs should be duplicated first and then assigned to picking lines or the other way round. Question 3 determines how the SKUs are assigned or clustered to the picking lines. The final question only comes into question when the SKUs are duplicated before assigning them to the picking lines and determines if the non-duplicate and duplicate SKUs are treated separately (ND) or simultaneously (S) when they are assigned to the picking lines.*

while others only require 3 decisions. Table 6.1 gives a break down of the algorithms discussed in this chapter based on the decision flow illustrated in Figure 6.1.

The first question is whether the original picking lines that are selected are to be treated separately (PS) or whether they will be combined into a single DBN list (PC). When they are treated separately the picking lines received from PEP are read into the algorithm and adjusted by means of duplicating and removing SKUs and assigning the removed SKUs to additional picking lines. While combining the original picking lines selected into a single DBN list will ignore the work previously done by PEPs employees at the DC when assigning the SKUs to picking lines.

The next question to consider is whether to duplicate the SKUs and then assign them to the various picking lines (DA) or to first assign them to the picking lines and then duplicate the SKUs on each of the picking lines in order for the SKUs to fill all the available locations on the picking lines (AD). This question only needs to be answered when it was decided to combine the original picking lines into a single DBN list, because when the original picking lines are kept separate they are already assigned to a picking line and therefore the SKUs can only be duplicated. When duplicating the SKUs on the combined DBN list before assigning them to picking lines, a number of SKUs equivalent to the total number of locations available on all the new picking lines less the number of unique SKUs on the combined DBN list, is to

Section	Algorithm	Decision flow
6.2.1	1	PS/D/M1-ND
6.2.2	2	PS/D/M1-S
6.3.1	3	PC/D/M1-ND
6.3.2	4	PC/D/M1-S
6.3.3	5	PC/D/M2-S
6.3.4	6	PC/D/M3-S
6.4.1	7	PC/M1/D
6.4.2	8	PC/M2/D
6.4.3	9	PC/M3/D
6.5.1	10	PC/C1/D
6.5.2	11	PC/C2/D
6.5.3	12	PC/C3/D
6.5.4	13	PC/C4/D
6.5.5	14	PC/C5/D
6.5.6	15	PC/C6/D

Table 6.1: *Break down of the algorithms discussed in the remainder of this chapter by the decision flow followed as illustrated in Figure 6.1.*

be duplicated. The SKUs are then assigned to the new picking lines by means of one of three assignment methods. This can cause an unequal number of SKUs being assigned to each of the new picking lines, but all picking lines are filled to an equal capacity. The alternative is to assign the SKUs to the new picking lines first by means of one of three assignment methods and then duplicating an approximately equal number of SKUs on each of the new picking lines in order for the SKUs to fill all the available locations on each of the new picking lines. This will lead to assigning approximately an equal number of SKUs to each of the new picking lines and therefore almost an equal number of duplicated SKUs will be present on each of the new picking lines.

The third question is to determine how the SKUs are going to be assigned to the new picking lines or the additional picking lines. There are three main assignment methods applied in this study. The SKUs are always assigned based on their pick frequency per location. This is the pick frequency of the SKU divided by the number of locations that will be assigned to the SKU. If the SKUs are first assigned to the picking lines and only duplicated at the end, the pick frequency per location is equal to the pick frequency of the SKU as each of the SKUs will occupy a single location at the time when it is assigned to a picking line. When the SKUs are, however, duplicated first, it is known how many locations a SKU will occupy and therefore the pick frequency per location is used instead. The first assignment method (M1) is to cyclical assign the SKUs to the new picking lines starting with the SKU with the highest pick frequency per location. This implies that the first SKU is assigned to the first picking line and the second SKU to the second picking line and so forth until each of the new picking lines have received one SKU and then assigning a second SKU to each of the picking lines and so forth until all the SKUs are assigned to a picking line and all the picking lines have reached their capacity. The second assignment method (M2), set length subset sequential assignment (SLSS), entails that the SKUs are assigned to the first picking line starting from the SKU with the highest pick frequency per location until the picking line has reached its capacity and only then starting to assign SKUs to the next picking line. The final assignment method (M3), remaining high-low cyclical assignment (RHLC), assigns the SKUs with the highest and lowest pick frequency per location simultaneously to the picking lines in a cyclical manner. Figure 6.2 illustrates how a set of 24 SKUs are assigned to 4 picking lines if each of the 3 assignment methods are implemented and SKU 1 is the SKU with the highest pick frequency per location and SKU 24 the SKU with

Picking line	Cyclical (M1)					
A	1	5	9	13	17	21
B	2	6	10	14	18	22
C	3	7	11	15	19	23
D	4	8	12	16	20	24

Picking line	SLSS (M2)					
A	1	2	3	4	5	6
B	7	8	9	10	11	12
C	13	14	15	16	17	18
D	19	20	21	22	23	24

Picking line	RHLC (M3)					
A	1	24	5	20	9	16
B	2	23	6	19	10	15
C	3	22	7	18	11	14
D	4	21	8	17	12	13

Figure 6.2: *An example of how a set of 24 SKUs are assigned to 4 picking lines if each of the 3 assignment methods are implemented and SKU 1 is the SKU with the highest pick frequency per location and SKU 24 the SKU with the lowest pick frequency per location.*

the lowest pick frequency per location.

The final question is only considered if the SKUs are duplicated first before assigning them to the various new picking lines. The question is whether to treat the non-duplicated and duplicated SKUs separate (ND) or simultaneously (S). By treating the duplicated and non-duplicated SKUs separately, the SKUs with the highest pick frequency on each of the picking lines are duplicated first when the original picking lines are treated separately and the non-duplicated SKUs with the highest pick frequencies on each of the original picking lines are removed and assigned to the additional picking lines. When the original picking lines are combined into a single DBN list the duplicated SKUs are assigned to the new picking lines by means of one of the assignment methods and only then are the non-duplicated SKUs assigned to the new picking lines by means of the same assignment method. If the duplicated and non-duplicated SKUs are treated simultaneously the SKUs are sorted in descending order of pick frequency per location on each of the original picking lines when they are treated separately and the SKUs with the highest pick frequency per locations will be assigned to the additional picking lines by means of one of the assignment methods. While for the instances where the original picking lines are first combined into a single DBN list the SKUs are assigned to the picking lines by means of one of the assignment methods until the SKUs assigned to each picking line will fill all the available locations.

In addition to the three assignment methods there are also 6 clustering methods. These clustering methods assign SKUs to the various picking lines first until all the SKUs have been assigned to a picking line and each picking line has reached its capacity. An approximately equal number of SKUs are therefore assigned to each of the picking lines. Only after the SKUs have been assigned to the picking lines are the SKUs with the highest pick frequency on each of the picking lines duplicated in order for the SKUs to be assigned to all the available locations.

All the algorithms constructed from this decision flow yield an ouptput file indicating which SKUs are assigned to which picking line if L picking lines are constructed. The output file thus contains set $\mathcal{S}^* = \{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_M, \ldots, \mathcal{S}_L\}$ containing sets of SKUs assigned to each picking line and a set indicating the number of locations each SKU in set $\mathcal{S}_k$ will occupy, $\mathcal{L}^* = \{\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_M, \ldots, \mathcal{L}_L\}$ as well as a set containing the pick frequency of the SKUs assigned to picking line k, $\mathcal{P}^* = \{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_M, \ldots, \mathcal{P}_L\}$.

6.2 Original picking lines, duplicate first, assign second

The first two algorithms is based on the fact that PEP already assigned DBNs to picking lines and therefore these original picking lines, as received from PEP, are used as a benchmark to measure the proposed algorithms. The algorithms described in §6.2.1 and §6.2.2 treat these original picking lines as the basis and duplicate and remove SKUs from these original picking lines. These two algorithms can be broken down into four steps.

- Step 1: Identify the SKUs with the highest pick frequencies and duplicate them.
- Step 2: Remove the excess SKUs from the picking lines.
- Step 3: Assign the removed SKUs to a new picking line.
- Step 4: Identify the non-duplicated SKUs with the highest pick frequencies on the additional picking lines and duplicate them until there are no unused locations on the additional picking lines.

As discussed in Chapter 3, PEP construct their picking lines by sorting the SKUs from highest pick quantity to lowest (not pick frequency) and spread them across the various picking lines so that they all have a more or less equal number of picks (total number of units picked), which is perceived as an equal distribution of work across the picking lines. This is of high importance to PEP due to the fact that pickers have the tendency to stay away from work if they know they are working on a "hard" picking line, in other words a picking line with a higher number of picks than the other picking lines.

The decision flow followed for Algorithms 1 and 2 are illustrated in Figure 6.3. Algorithms 1 and 2 treat the original picking lines selected separately and the SKUs with the highest pick frequency are duplicated. The SKUs that are removed from the original picking lines in order to open up space for the duplicated SKUs are assigned to the additional picking lines by means of assignent method 1. Algorithm 1 treat the non-duplicated and duplicated SKUs separately when SKUs are removed giving a decision flow of PS/D/M1-ND, while Algorithm 2 treat them simultaneously when SKUs are removed giving a decision flow of PS/D/M1-S.

The input file for Algorithm 1 and 2 is the same and contains

- the number of original picking lines selected (M),
- the total number of picking lines to construct (L),
- the total capacity of each of the picking lines (C, the total capacity of each picking line is set to 56 as all of PEP's picking lines contain 56 locations),
- the number of locations that the SKUs removed from the original picking line k needs to require (r_k),
- a set of SKUs $\mathcal{S}' = \{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_m, \ldots, \mathcal{S}_M\}$ containing sets of unique SKUs ($\mathcal{S}_m = \{s_{m1}, s_{m2}, \ldots, s_{mi}, \ldots, s_{mn}\}$) assigned to the original picking line m,
- a set indicating the number of locations each SKU in set $\mathcal{S}_m$ will occupy, $\mathcal{L}' = \{\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_m, \ldots, \mathcal{L}_L\}$ ($\mathcal{L}_m = \{l_{m1}, l_{m2}, \ldots, l_{mi}, \ldots, l_{mn}\}$),

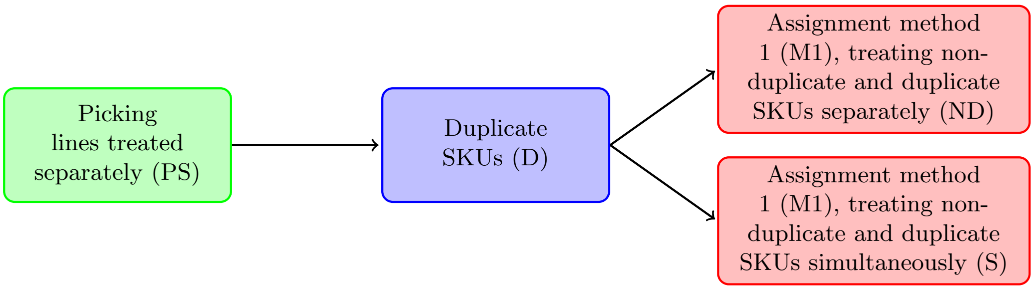


Figure 6.3: *A schematical representation of the decision flow when constructing Algorithms 1 and 2. Algorithm 1 and 2 treat the original picking lines selected separately and the SKUs with the highest pick frequency are duplicated, afterwhich the SKUs that are removed from the original picking lines in order to make space available for the duplicated SKUs are assigned to the additional picking lines by means of assignent method 1. Algorithm 1 treat the non-duplicated and duplicated SKUs separately when SKUs are removed giving a decision flow of PS/D/M1-ND, while Algorithm 2 treat them simultaneously when SKUs are removed giving a decision flow of PS/D/M1-S.*

- a set containing the pick frequency of the SKUs in the original picking line m, $\mathcal{P}' = \{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_m, \ldots, \mathcal{P}_L\}$ ($\mathcal{P}_m = \{p_{m1}, p_{m2}, \ldots, p_{mi}, \ldots, p_{mn}\}$).

Initially all SKUs only require a single location on a picking line, thus $l_{mi} = 1$ for every SKU i in set $\mathcal{S}_m$.

Algorithm 1 differ from Algorithm 2 in the sense that Algorithm 1 treat non-duplicate and duplicate SKUs separately when SKUs are removed from the original picking lines while Algorithm 2 remove both non-duplicate and duplicated SKUs from the original picking lines that are selected. The result hereof is that the number of SKUs that are duplicated on picking line m (d_m) in Algorithm 1 is not necessarily equal to the number of SKUs duplicated in Algorithm 2 and that the number of locations required by the SKUs removed from picking line m (r_m) might also be unequal for the 2 algorithms. The algorithm being implemented, number of original picking lines selected and the number of additional picking lines will determine the number of SKUs with the highest pick frequency that will be duplicated. Not all the original picking lines contained 56 SKUs as shown in Table 5.1. For these original picking lines additional SKUs are duplicated equal to the number of empty locations on the original picking lines.

For both algorithms the SKUs removed from the original picking lines are assigned to additional picking lines. If only one additional picking line is constructed then all the removed SKUs is assigned to the picking line. If more than one additional picking line is created, the removed SKUs from all the selected original picking lines are assigned to the various additional picking lines by cyclically assigning them to the picking lines ($k = M+1, M+2, \ldots, L$), asssignment Method 1. The same applies for the corresponding pick frequency and location entries of the removed SKUs. Thus if SKU s_{mj} is removed from picking line m and assigned to picking line k where $k = M+1, M+2, \ldots, L$ (in other words removed from set $\mathcal{S}_m$ and assigned to set $\mathcal{S}_k$), then l_{mj} is removed from set $\mathcal{L}_m$ and assigned to set $\mathcal{L}_k$ and p_{mj} is removed from set $\mathcal{P}_m$ and assigned to set $\mathcal{P}_k$.

The final step for Algorithm 1 and Algorithm 2 is to determine if there are any unused locations on the additional picking lines ($k = M+1, \ldots, L$). If this is the case the non-duplicate SKUs with the highest pick frequency per location is duplicated until all the available locations on the additional picking lines will be occupied by a SKU. For non-duplicate SKUs the pick frequency per location is equal to the SKUs' pick frequency as the SKUs are only assigned to a single location. Thus for the picking lines where $M < k \leq L$ the number of SKUs to duplicate is

$$\begin{aligned} d_k &= C - |\mathcal{L}_k|_\Sigma \\ &\approx C - \left\lfloor \frac{\sum_{m=1}^{M} r_m}{L-M} \right\rfloor \end{aligned}$$

and the number of SKUs to remove r_k is 0, where $|\mathcal{L}_k|_\Sigma = \sum_{i=1}^{n} l_i$ is the total number of locations recuired to accommodate the SKUs assigned to picking line k. The algorithm therefore determine the d_k non-duplicate SKUs on the additional picking line k ($M < k \leq L$) with the highest pick frequency and allocate it to 2 locations ($l_j \leftarrow 2$) on picking line k until the picking line has reached its capacity C.

6.2.1 Algorithm 1: PS/D/M1-ND

Algorithm 1 follow the decision flow of PS/D/M1-ND as discussed in §6.1 and illustrated in Figure 6.3 and can be summarized as duplicating the SKUs with the highest pick frequencies and then removing the non-duplicate SKUs with the highest pick frequencies. When duplicating the SKUs with the highest pick frequency on a picking line, a number of SKUs need to be removed from the picking line as well so that the number of SKUs on a picking line never exceed 56, as this is the total number of locations available on a picking line.

The d_m SKUs with highest pick frequency on the original picking line m ($m = 1, 2, \ldots, M$) is duplicated in step 1 of Algorithm 1, while step 2 determines that the r_m non-duplicate SKUs with the highest pick frequency are removed from picking line m. The number of SKUs to duplicate on the original picking lines, where $k = m$ for $k \leq M$ if all the original picking lines selected contains 56 SKUs, is

$$d_k \approx \left\lfloor \frac{(L-M)C}{L} \right\rfloor,$$

and the number of SKUs to remove on picking line k is $r_k = d_k$.

For Algorithm 1, the SKUs that were removed from the original picking lines are assigned to the additional picking lines in step 3 so that each additional picking line receives an approximately equal number of SKUs. The SKUs assigned to the additional picking lines is approximately equal to

$$|\mathcal{L}_k|_\Sigma \approx \left\lceil \frac{\sum_{m=1}^{M} r_m}{L-M} \right\rceil,$$

leaving locations open which are to be filled by duplicating SKUs on the additional picking lines. The final step of Algorithm 1 determines which of the additional picking lines require how many non-duplicated SKUs to be duplicated once SKUs have been duplicated and removed from all of the M original picking lines and assigned to the additional picking lines $k = M+1, M+2, \ldots, L$.

For Algorithm 1 the additional picking lines will receive no duplicated SKUs from the original picking lines. Thus an unequal number of duplications may occur on the original picking

Original picking line	New picking lines			
number	Picking line number	# SKUs	Duplications	Removals
1	1	42	14	14
2	2	42	14	14
3	3	42	14	14
–	4	42	14	0
1	1	34	22	22
2	2	34	22	22
3	3	34	22	22
–	4	33	23	0
–	5	33	23	0
1	1	29	27	27
2	2	29	27	27
3	3	29	27	27
–	4	27	27	0
–	5	27	27	0
–	6	27	27	0

Table 6.2: *Number of SKUs to duplicate and remove per picking line when creating additional lines from 3 original picking lines for Algorithm 1, if all the selected picking lines contain 56 SKUs.*

lines mainly because not all the original picking lines contained 56 SKUs initially, but the number of SKUs removed from each picking line is the same. Thus the number of SKUs removed will never exceed the number of SKUs duplicated from the original picking line. The SKUs that are not duplicated or removed from the original picking lines remain in their original picking line.

The number of SKUs to be duplicated on each picking line and removed from each picking line are specified as in Table 6.2 when 3 of the original picking lines are selected (in Appendix A additional tables are provided for when 2, 4 and 5 of the original picking lines are selected), if all the selected data sets contain 56 SKUs. The number of original picking lines selected and the number of SKUs on each of these original picking lines therefore determine the total number of SKUs to duplicate and remove from each data set. For instance, when selecting the original picking lines C, H and I, the original picking lines C and H need to be filled first as they contain 51 and 54 SKUs respectively. Thus for the original picking line C, 5 SKUs will need to be duplicated to fill the picking line and for the original picking line H, 2 SKUs. The original picking line I already contain 56 SKUs, therefore does not need any duplications to fill the picking line. When creating 4 picking lines from the 3 original picking lines, an additional 56 duplications is required so that all 4 lines have SKUs assigned to the available locations and none is left empty. Therefore on each of the original picking lines selected there need to be 14 SKUs duplicated so that the 14 SKUs can be removed and assigned to the new picking line. The 14 SKUs with the highest pick frequency on the new picking line also need to be duplicated to fill the new picking line. The SKUs duplicated and removed is shown in Table 6.3. SKUs are numbered in descending pick frequency, thus SKU 1 is the SKU with the highest pick frequency and SKU 56 the SKU with the lowest pick frequency for a specified picking line. The pseudo code for this algorithm is given in Algorithm 1.

6.2.2 Algorithm 2: PS/D/M1-S

The only difference between Algorithm 1 and Algorithm 2 is that where Algorithm 1 treat non-duplicate and duplicate SKUs separately when removing SKUs from the original picking lines

Original picking line	Picking line nr	SKUs duplicated	SKUs moved to new line
C	1	1 to 19	20 to 33
H	2	1 to 16	17 to 30
I	3	1 to 14	15 to 28
New line	4	1 to 14	n/a

Table 6.3: *Number of SKUs duplicated and removed on the original picking lines if selecting picking lines C, H and I to create 4 picking lines by means of Algorithm 1.*

Algorithm 1: PS/D/M1-ND

Input : The variables M, L, C and r_k and sets $\mathcal{S}', \mathcal{L}'$ and $\mathcal{P}'$.
Output : Sets $\mathcal{S}^*, \mathcal{L}^*$ and $\mathcal{P}^*$.

for $m = 1$ *to* M **do**
    $j \leftarrow 0$;
    $d \leftarrow 0$;
    $r \leftarrow 0$;
    Sort the SKUs in set $\mathcal{S}_m$ in descending order by pick frequency $p_{m,i}$ for the original picking line m;
    while $j < C$ **do**
        while $d < d_m$ **do**
            $j = j + 1$;
            $l_{m,j} \leftarrow 2$;
            $d = d + 1$;
        end
        while $r + l_{m,(j+1)} \leq r_m$ **do**
            for $k = M + 1$ *to* L **do**
                $j = j + 1$;
                $\mathcal{S}_k \leftarrow \mathcal{S}_k \cup s_{m,j}$;
                $\mathcal{L}_k \leftarrow \mathcal{L}_k \cup l_{m,j}$;
                $\mathcal{P}_k \leftarrow \mathcal{P}_k \cup p_{m,j}$;
                $\mathcal{S}_m \leftarrow \mathcal{S}_m \setminus s_{m,j}$;
                $\mathcal{L}_m \leftarrow \mathcal{L}_m \setminus l_{m,j}$;
                $\mathcal{P}_m \leftarrow \mathcal{P}_m \setminus p_{m,j}$;
                $r = r + l_{m,j}$;
            end
        end
    end
end
for $k = M + 1$ *to* L **do**
    if $C > |\mathcal{L}_k|_\Sigma$ **then**
        Sort SKUs in set $\mathcal{S}_k$ from highest pick frequency to lowest for picking line k;
        $j \leftarrow 1$;
        while $C > |\mathcal{L}_k|_\Sigma$ **do**
            $l_{k,j} \leftarrow 2$;
            $j = j + 1$;
        end
    end
end
$\mathcal{S}^* \leftarrow \mathcal{S}_1 \cup \mathcal{S}_2 \cup \ldots \cup \mathcal{S}_M \cup \mathcal{S}_{M+1} \cup \ldots \cup \mathcal{S}_L$;
$\mathcal{L}^* \leftarrow \mathcal{L}_1 \cup \mathcal{L}_2 \cup \ldots \cup \mathcal{L}_M \cup \mathcal{L}_{M+1} \cup \ldots \cup \mathcal{L}_L$;
$\mathcal{P}^* \leftarrow \mathcal{P}_1 \cup \mathcal{P}_2 \cup \ldots \cup \mathcal{P}_M \cup \mathcal{P}_{M+1} \cup \ldots \cup \mathcal{P}_L$;

($k = m = 1, 2, \ldots, M$) selected, Algorithm 2 treat them simultaneously. Thus for Algorithm 2 both duplicated and non-duplicated SKUs can be removed from the original picking lines to be assigned to the additional picking lines ($k = M + 1, M + 2, \ldots, L$). The decision flow of Algorithm 2 is therefore PS/D/M1-S as discussed in §6.1 and illustrated in Figure 6.3.

The first step of Algorithm 2 is the same as for Algorithm 1. While step 2 differs slightly from Algorithm 1 in the sence that the r_m SKUs with the highest pick frequency per location is removed from the original picking line m, where for Algorithm 1 the non-duplicate SKUs with the highest pick frequency is removed. The number of SKUs to duplicate on the original picking lines for Algorithm 2, where $k = m$ for $k \leq M$, if all the original picking lines selected contains 56 SKUs, is

$$d_k \approx \left\lfloor \frac{(L-M)C}{M} \right\rfloor,$$

while the number of SKUs to remove on picking line k is given by $r_k = d_k$, also slightly differ from Algorithm 1.

As for Algorithm 1, all the SKUs that are removed from the original picking lines are assigned to the additional picking lines by cyclically assigning the SKUs to the additional picking lines ($k = M+1, M+2, \ldots, L$). The total number of SKUs to be assigned to each of the additional picking lines from all of the original picking lines is split more or less equal across the additional picking lines, where the SKUs assigned to the additional picking lines is calculated as for Algorithm 1.

As both duplicated and non-duplicated SKUs are removed from picking lines 1 to M, there is no need to make provision to duplicate additional non-duplicate SKUs on the picking lines $M+1$ to L, as in the final step of Algorithm 1. However, the algorithm still needs to check that all the new picking lines contain SKUs that will fill the 56 available locations on each picking line because when removing SKUs that require an equal number of locations from each of the original picking lines, it is possible that when these SKUs are distributed over the additional picking lines that they will not fill all the available locations on each picking line. This will occur if 5 of the original picking lines are selected, and to remove SKUs that will require an equal number of locations from each picking line will not fill the 56 locations available on the additional picking line, an additional non-duplicated SKU on the additional picking line need to be duplicated so that all the locations gets a SKU assigned to it.

The final step of Algorithm 2 is thus the same as for Algorithm 1 and determines which of the additional picking lines require how many non-duplicated SKUs to be duplicated once SKUs have been duplicated and removed from all of the M original picking lines and assigned to the additional picking lines $k = M+1, M+2, \ldots, L$. Table 6.4 indicates the number of SKUs to duplicate and remove on each picking line if 3 of the original picking lines are selected and contains 56 SKUs. In Appendix A additional tables are provided for when 2, 4 and 5 of the original picking lines are selected. The pseudo code is given in Algorithm 2.

6.3 Combined DBN list, duplicate SKUs first, assign second

The algorithms described in the previous section require that the SKUs are assigned to a picking line igoring how PEP historically did it. By combining the original picking lines into a single DBN list, the possibilities when constructing the picking lines is increased and therefore improved picking lines are more likely. Thus for all the algorithms in the remainder of this chapter, the original picking lines are combined into a single DBN list first, as if they have not been assigned at all.

Algorithm 2: PS/D/M1-S

Input : The variables M, L, C and r_k and sets $\mathcal{S}', \mathcal{L}'$ and $\mathcal{P}'$.
Output : Sets $\mathcal{S}^*, \mathcal{L}^*$ and $\mathcal{P}^*$.

$m \leftarrow 1$;
while $m \leq M$ **do**
  $j \leftarrow 0$;
  $d \leftarrow 0$;
  $r \leftarrow 0$;
  Sort the SKUs in set $\mathcal{S}_m$ in descending order by pick frequency $p_{m,i}$ for the original picking line m;
  while $j \leq C$ **do**
    while $d < d_m$ **do**
      $l_j \leftarrow 2$;
      $j = j + 1$;
      $d = d + 1$;
    end
    Sort the SKUs in set $\mathcal{S}_m$ in descending order by pick frequency per location ($\frac{p_i}{l_i}$);
    for $k = M + 1$ *to* L **do**
      while $r + l_{m,(j+1)} \leq r_m$ **do**
        $j = j + 1$;
        $\mathcal{S}_k \leftarrow \mathcal{S}_k \cup s_{m,j}$;
        $\mathcal{L}_k \leftarrow \mathcal{L}_k \cup l_{m,j}$;
        $\mathcal{P}_k \leftarrow \mathcal{P}_k \cup p_{m,j}$;
        $\mathcal{S}_m \leftarrow \mathcal{S}_m \setminus s_{m,j}$;
        $\mathcal{L}_m \leftarrow \mathcal{L}_m \setminus l_{m,j}$;
        $\mathcal{P}_m \leftarrow \mathcal{P}_m \setminus p_{m,j}$;
        $r = r + l_{m,j}$;
      end
    end
  end
  $m = m + 1$;
end
for $k = M + 1$ *to* L **do**
  if $C > |\mathcal{L}_k|_\Sigma$ **then**
    Sort SKUs in set $\mathcal{S}_k$ from highest pick frequency to lowest for picking line k;
    $j \leftarrow 1$;
    while $C > |\mathcal{L}_k|_\Sigma$ **do**
      $l_{k,j} \leftarrow 2$;
      $j = j + 1$;
    end
  end
end
$\mathcal{S}^* \leftarrow \mathcal{S}_1 \cup \mathcal{S}_2 \cup \ldots \cup \mathcal{S}_M \cup \mathcal{S}_{M+1} \cup \ldots \cup \mathcal{S}_L$;
$\mathcal{L}^* \leftarrow \mathcal{L}_1 \cup \mathcal{L}_2 \cup \ldots \cup \mathcal{L}_M \cup \mathcal{L}_{M+1} \cup \ldots \cup \mathcal{L}_L$;
$\mathcal{P}^* \leftarrow \mathcal{P}_1 \cup \mathcal{P}_2 \cup \ldots \cup \mathcal{P}_M \cup \mathcal{P}_{M+1} \cup \ldots \cup \mathcal{P}_L$;

The decision flow followed for Algorithms 3–6 are illustrated in Figure 6.4. Algorithms 3–6 combine the original picking lines selected and duplicate a sufficient number of SKUs with the highest pick frequency, then the SKUs are assigned to the picking lines by means of one of the assignent methods. Algorithm 3 treat the non-duplicated and duplicated SKUs separately when SKUs are assigned to the new picking lines giving a decision flow of PC/D/M1-ND, while Algorithms 4–6 treat them simultaneously when SKUs are assigned to the new picking lines giving a decision flow of PC/D/M1-S, PC/D/M2-S and PC/D/M3-S respectively.

The input file for the algorithms discussed in this section vary from that in §6.2 and consists of

Original picking line number	New picking lines		
	Picking line number	Duplications	Removals
1	1	18	18
2	2	18	18
3	3	18	18
–	4	2	0
1	1	36	36
2	2	36	36
3	3	36	36
–	4	2	0
–	5	2	0
1	1	54	54
2	2	54	54
3	3	54	54
–	4	2	0
–	5	2	0
–	6	2	0

Table 6.4: *Number of SKUs to duplicate and remove per picking line when constructing new picking lines when selecting 3 of the original picking lines for Algorithm 2, if all of the selected original picking lines contained 56 SKUs.*

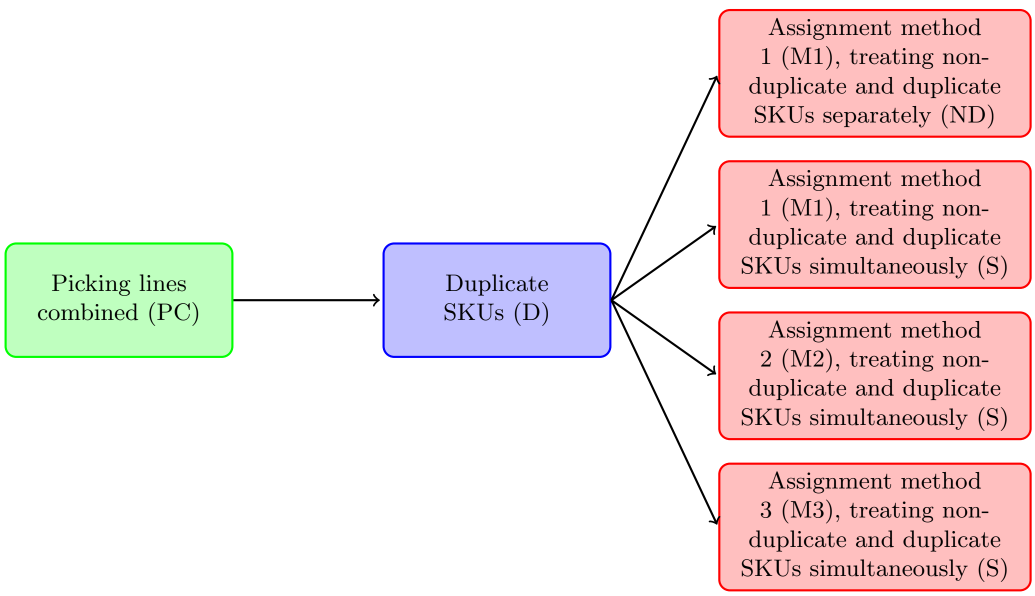


Figure 6.4: *A schematical representation of the decision flow when constructing Algorithms 3–6. Algorithms 3–6 combine the original picking lines selected and then duplicate the SKUs with the highest pick frequency, then the SKUs are assigned to the new picking lines by means of one of the assignent methods. Algorithm 3 treat the non-duplicated and duplicated SKUs separately when SKUs are assigned to the new picking lines giving a decision flow of PC/D/M1-ND, while Algorithms 4–6 treat them simultaneously when SKUs are assigned to the new picking lines giving a decision flow of PC/D/M1-S, PC/D/M2-S and PC/D/M3-S respectively.*

- a set of SKUs ($\mathcal{S} = \{s_1, s_2, \ldots, s_i, \ldots, s_n\}$) that need to be assigned to various picking lines (L) containing only non-duplicate SKUs,
- the pick frequency for each SKU is given by $\mathcal{P} = \{p_1, p_2, \ldots, p_i, \ldots, p_n\}$,
- the number of locations that each SKU is to allocated to, $\mathcal{L} = \{l_1, l_2, \ldots, l_i, \ldots, l_n\}$,
- the capacity of each picking line (C), indicating the number of locations the SKUs can be assigned to for each picking line and
- the number of picking lines (L) to construct

Initially all SKUs only require a single location on a picking line, thus $l_i = 1$ for every SKU i in set $\mathcal{S}$.

As mentioned, for the algorithms disdussed in this section, the SKUs with the highest pick frequencies are duplicated before assigning the SKUs to the new picking lines. Chapter 5 clarified the reason for duplicating the SKUs with the highest pick frequency first. The number of SKUs to duplicate are determined by the number of SKUs in set $\mathcal{S}$ ($|\mathcal{L}|_\Sigma$) and the number of locations available on L picking lines (CL). Table 6.5 indicate the number of SKUs to be duplicated on the combined DBN list (set $\mathcal{S}$) before assigning them to picking lines, for set $\mathcal{S}$ to fill the CL locations available in L picking lines (if all the original picking lines included in set $\mathcal{S}$ contains 56 SKUs). For instances where the original picking lines included in set $\mathcal{S}$ does not contain 56 SKUs each, additional SKUs need to be duplicated equal to the number of SKUs that was short on the original picking lines so that all the locations in the new picking lines contain a SKU and none is left unused. Therefore the total SKUs duplicated is equal to $CL - |\mathcal{L}|_\Sigma$. If SKU j in set $\mathcal{S}$ is duplicated, then $l_j = 2$.

The third decision, as discussed in §6.1, is if the non-duplicate and duplicate SKUs are to be treated separately when they are being assigned to the various picking lines or whether they should be assigned simultaneously. By treating them separately the duplicated SKUs are assigned to the picking lines by means of one of the assignment methods starting with the SKU with the highest pick frequency and then the non-duplicate SKUs are assigned to the picking lines by means of the same assignment method as the duplicated SKUs. Because the duplicate and non-duplicate SKUs are treated separately, the order in which the SKUs will be assigned to the picking lines is the same whether they are assigned from the highest pick frequency (p_i) to the lowest or from the highest pick frequency per location ($\frac{p_i}{l_i}$) to the lowest. When the non-duplicate and duplicate SKUs are assigned to the picking lines simultaneously, they are first sorted in descending order of the pick frequency per location ($\frac{p_i}{l_i}$) before assigning them to the picking lines. The duplicated SKUs' pick frequency per location is equal to half of its pick frequency as these SKUs are assigned to two locations on the picking lines.

The final step for the algorithms in this section is to assign the SKUs to the various picking lines by means of one of the 3 assignment methods discussed in §6.1. Algorithm 3 and 4 assign the SKUs by means of assignment method 1 by means of cyclical assignment. Algorithm 5 assign the SKUs to the picking lines by means of assignment method 2, SLSS assignment, Algorithm 6 assign the SKUs to the picking lines by means of assignment method 3, RHLC assignment.

Nr original picking lines (M)	Nr new picking lines (L)	$L - M$	Total duplications ($C(L-M)$)
2	3	1	56
2	4	2	112
3	4	1	56
3	5	2	112
3	6	3	168
4	5	1	56
4	6	2	112
4	7	3	168
4	8	4	224
5	6	1	56
5	7	2	112
5	8	3	168
5	9	4	224
5	10	5	280

Table 6.5: *Number of SKUs to duplicate when creating new picking lines from a number of selected original picking lines for Algorithms 3 – 4, if all the original picking lines contain 56 SKUs.*

6.3.1 Algorithm 3: PC/D/M1-ND

For Algorithm 3 the SKUs that are to be duplicated are treated seperately from those that will only be assigned to a single location on the picking lines. The algorithm can therefore be split into two parts, the first will identify which SKUs to duplicate and assign them to the picking lines and the second part will assign the remaining non-duplicated SKUs to the picking lines.

The SKU with the highest pick frequency (p_i) in set $\mathcal{S}$ is duplicated by assigning $l_i = 2$ and assign it to picking line k. The SKUs are cyclically assigned to the various picking lines as was the case for the algorithms described in §6.2. Once a sufficient number of SKUs have been duplicated, the remainder of the SKUs will only be allocated to a single location and are therefore merely be cyclically assigned to the various picking lines. The pseudo code for this algorithm is given in Algorithm 3.

As the duplicated and non-duplicated SKUs are assigned to the picking lines seperately, there are more or less an equal number of duplicated and non-duplicated SKUs on each picking line. Thus the number of duplicated SKUs on each picking line k is approximately

$$\left\lceil \frac{CL - |\mathcal{L}_k|_\Sigma}{L} \right\rceil,$$

while the number of non-duplicated SKUs on each picking line is approximately

$$C - 2\left\lceil \frac{CL - |\mathcal{L}_k|_\Sigma}{L} \right\rceil.$$

6.3.2 Algorithm 4: PC/D/M1-S

For Algorithm 4, instead of identifying which SKUs are to be duplicated and assigning them to the picking lines first as for Algorithm 3 in §6.3.1, the SKUs that are to be duplicated and the non-duplicated SKUs are assigned to the picking lines simultaneously. This requires that the pick frequency per location is calculated instead of the total pick frequency per SKU as some

Algorithm 3: PC/D/M1-ND

Input : The variables L and C and sets $\mathcal{S}, \mathcal{L}$ and $\mathcal{P}$.
Output : Sets $\mathcal{S}^*, \mathcal{L}^*$ and $\mathcal{P}^*$.

Sort the SKUs in $\mathcal{S}$ from highest to lowest pick frequency (p_i);
$j \leftarrow 1$;
while $j < n$ **do**
  for $k = 1$ *to* L **do**
    if $|\mathcal{L}^*|_\Sigma \leq CL$ **then**
      $l_j \leftarrow 2$;
      $j = j + 1$;
      $\mathcal{S}_k \leftarrow \mathcal{S}_k \cup s_j$;
      $\mathcal{L}_k \leftarrow \mathcal{L}_k \cup l_j$;
      $\mathcal{P}_k \leftarrow \mathcal{P}_k \cup p_j$;
    end
    else if $|\mathcal{L}_k|_\Sigma + l_{j+1} \leq C$ **then**
      $j = j + 1$;
      $\mathcal{S}_k \leftarrow \mathcal{S}_k \cup s_j$;
      $\mathcal{L}_k \leftarrow \mathcal{L}_k \cup l_j$;
      $\mathcal{P}_k \leftarrow \mathcal{P}_k \cup p_j$;
    end
  end
end
$\mathcal{S}^* \leftarrow \mathcal{S}_1 \cup \mathcal{S}_2 \cup \ldots \cup \mathcal{S}_L$;
$\mathcal{L}^* \leftarrow \mathcal{L}_1 \cup \mathcal{L}_2 \cup \ldots \cup \mathcal{L}_L$;
$\mathcal{P}^* \leftarrow \mathcal{P}_1 \cup \mathcal{P}_2 \cup \ldots \cup \mathcal{P}_L$;

SKUs will occupy two locations on a picking line. The algorithm can also be split into two parts, the first part will determine which SKUs to duplicate, whereas the second will assign the SKUs to picking lines by means of assignment method 1, cyclical assignment.

The second part considers the pick frequency per location ($\frac{p_i}{l_i}$) of each SKUs and sort the SKUs in set $\mathcal{S}$ in descending order. The SKUs are sorted by their pick frequency per location as the non-duplicate and duplicate SKUs are assigned to the picking lines simultaneously and not separately as for Algorithm 3. For Algorithm 4 the SKUs are assigned to the picking lines by means of assignment method 1 discussed in §6.1. The SKUs are thus cyclically assigned to the various picking lines as for Algorithm 3. When a picking line's capacity is reached ($|\mathcal{L}_k|_\Sigma = C$) before all SKUs have been assigned, the picking line is excluded from further assigning the remaining SKUs. Calculating the number of bin locations required per picking line after each assignment and not the number of SKUs assigned to the picking line, also ensures that the SKU and its duplicate is always on the same picking line and that the picking line's capacity is not exceeded. Due to the fact that the non-duplicate and duplicate SKUs are assigned to the picking lines simultaneously, there is an unequal number of duplicate and non-duplicate SKUs distributed over the picking lines.

6.3.3 Algorithm 5: PC/D/M2-S

This algorithm also consists of two parts of which the first is the same as for Algorithm 4, in the sense that the SKUs with the highest pick frequency in set $\mathcal{S}$ need to be identified and are to be duplicated and only then are the SKUs assigned to the various picking lines. Algorithm 5 only differs from Algorithm 4 in the method by which the SKUs are assigned to the picking lines.

Algorithm 4: PC/D/M1-S

Input : The variables L and C and sets $\mathcal{S}, \mathcal{L}$ and $\mathcal{P}$.
Output : Sets $\mathcal{S}^*, \mathcal{L}^*$ and $\mathcal{P}^*$.

Sort the SKUs in $\mathcal{S}$ from highest to lowest pick frequency (p_i);
$j \leftarrow 1$;
while $|\mathcal{L}|_\Sigma \leq CL$ **do**
    $l_j \leftarrow 2$;
    $j = j + 1$;
end
Sort the SKUs in $\mathcal{S}$ from highest to lowest pick frequency per location ($\frac{p_i}{l_i}$);
while $j < \frac{n}{2}$ **do**
    for $k = 1$ *to* L **do**
        if $|\mathcal{L}_k|_\Sigma + l_{j+1} \leq C$ **then**
            $j = j + 1$;
            $\mathcal{S}_k \leftarrow \mathcal{S}_k \cup s_j$;
            $\mathcal{L}_k \leftarrow \mathcal{L}_k \cup l_j$;
            $\mathcal{P}_k \leftarrow \mathcal{P}_k \cup p_j$;
        end
    end
end
$\mathcal{S}^* \leftarrow \mathcal{S}_1 \cup \mathcal{S}_2 \cup \ldots \cup \mathcal{S}_L$;
$\mathcal{L}^* \leftarrow \mathcal{L}_1 \cup \mathcal{L}_2 \cup \ldots \cup \mathcal{L}_L$;
$\mathcal{P}^* \leftarrow \mathcal{P}_1 \cup \mathcal{P}_2 \cup \ldots \cup \mathcal{P}_L$;

Algorithm 5 therefore also duplicate the SKUs before assigning them to picking lines. Once the total locations required by the SKUs in set $\mathcal{S}$ is equal to the locations available on the L picking lines, the SKUs are assigned to the picking lines from highest pick frequency per location ($\frac{p_i}{l_i}$) to the lowest and not from highest initial pick frequency (p_i) to lowest. The SKUs are assigned to a picking line k until picking line k reaches its capacity C and then SKUs are assigned to picking line $k+1$ and so forth. This determines that the SKUs with the highest frequency per location are grouped together on a picking lines and the next highest and so forth. Therefore instead of cyclically assigning the SKUs to the picking lines, the SKUs are assigned to the picking lines by means of SLSS assignment. In the instance where a picking line reaches its capacity before the other picking lines it is further excluded from the assigning of SKUs. The pseudo code for this algorithm is given in Algorithm 5.

The problem with this approach is that all the high pick frequency SKUs are more or less grouped together and all the low pick frequency SKUs are on the same picking line. This creates unequal work balance between the picking lines.

6.3.4 Algorithm 6: PC/D/M3-S

Algorithm 6 follows the same decision flow as Algorithms 4 and 5, but make use of assignment method 3 instead. It therefore also consists of two parts where the first part will determine which SKUs on the combined DBN list (set $\mathcal{S}$) to duplicate and the second part will be assigning the SKUs to the various picking lines. The first part of the algorithm is therefore exactly the same as the first part of Algorithm 4 and 5.

Once the total locations required by the SKUs in set $\mathcal{S}$ is equal to the locations available on the L picking lines, the SKUs are assigned to the picking lines by means of assignment method 3. Whereas for Algorithm 5 SKUs where assigned to picking lines by means of SLSS assignment, for Algorithm 6 they are assigned by means of RHLC assignment. Therefore the second part of

Algorithm 5: PC/D/M2-S

Input : The variables L and C and sets $\mathcal{S}, \mathcal{L}$ and $\mathcal{P}$.
Output : Sets $\mathcal{S}^*, \mathcal{L}^*$ and $\mathcal{P}^*$.

Sort the SKUs in $\mathcal{S}$ from highest to lowest pick frequency (p_i);
$j \leftarrow 1$;
while $|\mathcal{L}|_\Sigma \leq CL$ **do**
    $l_j \leftarrow 2$;
    $j = j + 1$;
end
Sort the SKUs in $\mathcal{S}$ from highest to lowest pick frequency per location ($\frac{p_i}{l_i}$);
$j \leftarrow 0$;
$k \leftarrow 1$;
while $j < n$ **do**
    while $|\mathcal{L}_k|_\Sigma + l_{j+1} \leq C$ **do**
        $j = j + 1$;
        $\mathcal{S}_k \leftarrow \mathcal{S}_k \cup s_j$;
        $\mathcal{L}_k \leftarrow \mathcal{L}_k \cup l_j$;
        $\mathcal{P}_k \leftarrow \mathcal{P}_k \cup p_j$;
    end
    $k = k + 1$;
end
$\mathcal{S}^* \leftarrow \mathcal{S}_1 \cup \mathcal{S}_2 \cup \ldots \cup \mathcal{S}_L$;
$\mathcal{L}^* \leftarrow \mathcal{L}_1 \cup \mathcal{L}_2 \cup \ldots \cup \mathcal{L}_L$;
$\mathcal{P}^* \leftarrow \mathcal{P}_1 \cup \mathcal{P}_2 \cup \ldots \cup \mathcal{P}_L$;

the algorithm is two fold, in the sense that the SKUs with the highest and lowest pick frequency per location needs to be determined simultaneously. The pick frequency per location of the duplicated SKUs ($l_i = 2$) are half of the original pick frequency for the SKU due to the fact that the SKU will be allocated to two locations on a picking line. For the duplicated SKUs both locations will be on the same picking line. In the instance where a picking line reaches its capacity before the other picking lines it is further excluded from the assigning of SKUs. The pseudo code for this algorithm is given in Algorithm 6.

6.4 Combined DBN list, assign SKUs first, duplicate second

In the previous section algorithms was described based on constructing L picking lines from a combined DBN list by first duplicating a sufficient number of SKUs ($CL - |\mathcal{L}|_\Sigma$) with the highest pick frequency and only then assigning the SKUs to the various picking lines based on the pick frequency per location ($\frac{p_i}{l_i}$). This section will discuss algorithms where L picking lines are to be constructed from a combined DBN list, from M original picking lines, by assigning the SKUs to the various picking lines based on the SKU pick frequency, by means of assignment methods 1–3 as for the algorithms discussed in §6.3. The SKUs with the highest pick frequencies on each new picking line are only duplicated after the SKUs have been assigned to the picking lines. Thus there are approximately the same number of duplicated SKUs on each of the picking lines. The decision flow followed for Algorithms 7–9 are illustrated in Figure 6.5, where Algorithm 7 follows a decision flow of PC/M1/D, Algorithm 8 follows PC/M2/D and Algorithm 9 follows PC/M3/D.

The input file for the algorithms discussed in this section is the same as for the algorithms in §6.3 with the addition of the number of SKUs assigned to each new picking line k which is indicated by c_k. The number of SKUs to be assigned to each picking line is split more or less

Algorithm 6: PC/D/M3-S

Input : The variables L and C and sets $\mathcal{S}, \mathcal{L}$ and $\mathcal{P}$.
Output : Sets $\mathcal{S}^*, \mathcal{L}^*$ and $\mathcal{P}^*$.
Sort the SKUs in $\mathcal{S}$ from highest to lowest pick frequency (p_i);
$j \leftarrow 1$;
while $|\mathcal{L}|_\Sigma \leq CL$ **do**
    $l_j \leftarrow 2$;
    $j = j + 1$;
end
Sort the SKUs in $\mathcal{S}$ from highest to lowest pick frequency per location ($\frac{p_i}{l_i}$);
$j \leftarrow 0$;
while $j < n$ **do**
    for $k = 1$ *to* L **do**
        if $|\mathcal{L}_k|_\Sigma + l_{j+1} \leq C$ **then**
            $j = j + 1$;
            $\mathcal{S}_k \leftarrow \mathcal{S}_k \cup s_j \cup s_{n-j+1}$;
            $\mathcal{L}_k \leftarrow \mathcal{L}_k \cup l_j \cup l_{n-j+1}$;
            $\mathcal{P}_k \leftarrow \mathcal{P}_k \cup p_j \cup p_{n-j+1}$;
        end
    end
end
$\mathcal{S}^* \leftarrow \mathcal{S}_1 \cup \mathcal{S}_2 \cup \ldots \cup \mathcal{S}_L$;
$\mathcal{L}^* \leftarrow \mathcal{L}_1 \cup \mathcal{L}_2 \cup \ldots \cup \mathcal{L}_L$;
$\mathcal{P}^* \leftarrow \mathcal{P}_1 \cup \mathcal{P}_2 \cup \ldots \cup \mathcal{P}_L$;

equal across the picking lines, leaving locations open which are to be filled by duplicating SKUs on the picking lines. Initially all SKUs only require a single location on a picking line, thus $l_i = 1$ for every SKU i in set $\mathcal{S}$.

The number of SKUs to assign to a picking line ($c_k \approx \lceil \frac{n}{L} \rceil$) for the algorithms discussed in this section, is determined by the number of picking lines to construct (L) and the number of SKUs in the original picking lines selected and is shown in Table 6.6 for when 3 of the original picking lines are selected and each of them contains 56 unique SKUs (additional tables are provided in Tables A.7–A.9, for when 2, 4 and 5 of the original picking lines are selected respectively). Each new picking line receives approximately an equal number of SKUs ($c_k \approx 56M/L$) and is filled in the second part of the algorithms by duplicating the SKUs with the highest pick frequency on each picking line so that all 56 locations on the picking line is occupied, therefore $c_k + d_k = C$. In some cases it is impossible to assign an equal number of SKUs to each new picking line, as in the case when 3 original picking lines is selected and 5 new picking lines is to be constructed. In this case 33.6 SKUs need to be assigned to each picking line for an equal distribution to occur, but as per the assumptions a SKU need to be assigned to a single picking line and cannot be distributed over multiple picking lines. Thus at least one picking line receives one more SKU. For example, when creating 5 new picking lines from 3 original picking lines, two lines will contain 33 SKUs and three lines will contain 34 SKUs.

The second part of the algorithms are exactly the same for all the algorithms in this section and determines whether the new sets of SKUs assigned to each picking line ($\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_k, \ldots, \mathcal{S}_L$) will fill them to capacity. The number of SKUs assigned to each of the new picking lines will be less than the number of locations available on each picking line, therefore it is required to duplicate SKUs in order to fill the picking line to capacity. The number of SKUs to duplicate

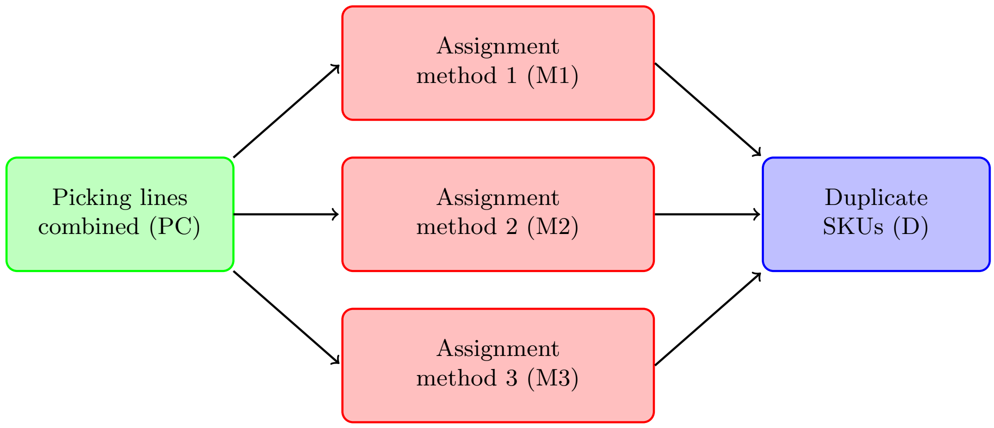


Figure 6.5: *A schematical representation of the decision flow when constructing Algorithms 7–9. Algorithms 7–9 combine the original picking lines selected and then SKUs are assigned to the new picking lines by means of one of the assignent methods, afterwhich the SKUs on each of the new picking lines with the highest pick frequencies are duplicated. Therefore the decision flow for Algorithm 7 is PC/M1/D, for Algorithm 8 is PC/M2/D and for Algorithm 9 is PC/M3/D.*

on picking line k is equal to

$$\begin{aligned} d_k &= C - c_k \\ &= C - |\mathcal{L}_k|_\Sigma. \end{aligned}$$

Therefore d_k will be greater than 0 if the SKUs assigned to picking line k require less than 56 locations, $C > |\mathcal{L}_k|_\Sigma$. The d_k SKUs on the new picking line k with the highest pick frequency (p_i) is therefore duplicated ($l_i \leftarrow 2$ for the relevant SKUs) in order to fill the new picking line so that none of the available locations are left unused as this is one of the assumptions discussed in Chapter 5. SKUs are thus duplicated until the number of locations required by the SKUs assigned to picking line k ($|\mathcal{L}_k|_\Sigma$) is equal to the capacity of the picking line ($C = 56$), thus when $d_k = 0$ and $C = |\mathcal{L}_k|_\Sigma$. Table 6.6 (as well as Tables A.7–A.9 for when 2, 4 and 5 of the original picking lines are selected) indicate the number of SKUs assigned to each of the new picking lines k (c_k) and the number of SKUs to duplicate for each picking line k if each of the M original picking lines selected contain 56 SKUs when 3 original picking lines are selected.

6.4.1 Algorithm 7: PC/M1/D

As for the algorithms in §6.3, the algorithms discussed in this section can also be split in two parts. Whereas in the previous section the first part consisted of identifying the SKUs with the highest pick frequency and duplicating them and the second part assigning the SKUs to the various picking line based on their pick frequency per bin location, the algorithms discussed in this section first assign the SKUs to the various picking lines based on their pick frequency whereafter the SKUs on the individual picking lines with the highest pick frequencies are duplicated. Algorithm 7 cyclically assign the SKUs from the combined set.

The first part assign the various SKUs in $\mathcal{S}$ to the L picking lines by means of assignment method 1, as for Algorithm 4. Once all the SKUs have been assigned to the L picking lines, the picking lines need to be filled by duplicating the SKUs with the highest pick frequency on each picking line. As only non-duplicate SKUs are assigned to the picking lines, the pick frequency of the SKU is equal to its pick frequency per location.

Nr picking lines (L)	Picking line (k)	#SKUs (c_k)	Duplications (d_k)
4	1	42	14
	2	42	14
	3	42	14
	4	42	14
5	1	34	22
	2	34	22
	3	34	22
	4	33	23
	5	33	23
6	1	28	28
	2	28	28
	3	28	28
	4	28	28
	5	28	28
	6	28	28

Table 6.6: *Number of SKUs to be assigned to each new picking line and the number of duplications required to fill all the locations on each line, when selecting 3 of the original picking lines and applying Algorithms 7–9, 10 and 11, if all the original picking lines contain 56 SKUs.*

Algorithm 7: PC/M1/D

Input : The variables L, C and c_k and sets $\mathcal{S}, \mathcal{L}$ and $\mathcal{P}$.
Output : Sets $\mathcal{S}^*, \mathcal{L}^*$ and $\mathcal{P}^*$.

Sort the SKUs in $\mathcal{S}$ from highest to lowest pick frequency (p_i);
$j \leftarrow 0$;
while $j < n$ **do**
  for $k = 1$ *to* L **do**
    if $|\mathcal{L}_k|_\Sigma + l_{j+1} \leq C$ **then**
      $j = j + 1$;
      $\mathcal{S}_k \leftarrow \mathcal{S}_k \cup s_j$;
      $\mathcal{L}_k \leftarrow \mathcal{L}_k \cup l_j$;
      $\mathcal{P}_k \leftarrow \mathcal{P}_k \cup p_j$;
    end
  end
end
for $k = 1$ *to* L **do**
  if $C > |\mathcal{L}_k|_\Sigma$ **then**
    Sort SKUs in set $\mathcal{S}_k$ from highest pick frequency per location to lowest for picking line k;
    $j \leftarrow 1$;
    while $C > |\mathcal{L}_k|_\Sigma$ **do**
      $l_j \leftarrow 2$;
      $j = j + 1$;
    end
  end
end
$\mathcal{S}^* \leftarrow \mathcal{S}_1 \cup \mathcal{S}_2 \cup \ldots \cup \mathcal{S}_L$;
$\mathcal{L}^* \leftarrow \mathcal{L}_1 \cup \mathcal{L}_2 \cup \ldots \cup \mathcal{L}_L$;
$\mathcal{P}^* \leftarrow \mathcal{P}_1 \cup \mathcal{P}_2 \cup \ldots \cup \mathcal{P}_L$;

6.4.2 Algorithm 8: PC/M2/D

As for Algorithm 7 in §6.4.1, Algorithm 8 can be split into two parts the first assigning c_k SKUs to the various picking lines and the second duplicating the d_k SKUs with the highest pick frequency on each new picking line k, as indicated in Table 6.6 for 3 original picking lines that each contains 56 unique SKUs (additional examples are provided in Tables A.7–A.9). Therefore determing the order in which SKUs are assigned to a picking line is the same as for Algorithm 7.

The first part assign the various SKUs in $\mathcal{S}$ to the L picking lines by means of assignment method 2, as for Algorithm 5. As for Algorithm 7 the d_k SKUs on the new picking line k with the highest pick frequency (p_i) is duplicated ($l_i \leftarrow 2$ for the relevant SKUs) in order to fill the new picking line so that none of the available locations are left unused as this is one of the assumptions discussed in Chapter 5.

Algorithm 8: PC/M2/D

Input : The variables L, C and c_k and sets $\mathcal{S}, \mathcal{L}$ and $\mathcal{P}$.
Output : Sets $\mathcal{S}^*, \mathcal{L}^*$ and $\mathcal{P}^*$.

Sort the SKUs in $\mathcal{S}$ from highest to lowest pick frequency (p_i);
$j \leftarrow 0$;
$k \leftarrow 1$;
while $j < n$ **do**
    while $|\mathcal{L}_k|_\Sigma + l_{j+1} \leq c_k$ **do**
        $j = j + 1$;
        $\mathcal{S}_k \leftarrow \mathcal{S}_k \cup s_j$;
        $\mathcal{L}_k \leftarrow \mathcal{L}_k \cup l_j$;
        $\mathcal{P}_k \leftarrow \mathcal{P}_k \cup p_j$;
    end
    $k = k + 1$;
end
for $k = 1$ *to* L **do**
    if $C > |\mathcal{L}_k|_\Sigma$ **then**
        Sort SKUs in set $\mathcal{S}_k$ from highest pick frequency per location to lowest for picking line k;
        $j \leftarrow 1$;
        while $C > |\mathcal{L}_k|_\Sigma$ **do**
            $l_j \leftarrow 2$;
            $j = j + 1$;
        end
    end
end
$\mathcal{S}^* \leftarrow \mathcal{S}_1 \cup \mathcal{S}_2 \cup \ldots \cup \mathcal{S}_L$;
$\mathcal{L}^* \leftarrow \mathcal{L}_1 \cup \mathcal{L}_2 \cup \ldots \cup \mathcal{L}_L$;
$\mathcal{P}^* \leftarrow \mathcal{P}_1 \cup \mathcal{P}_2 \cup \ldots \cup \mathcal{P}_L$;

6.4.3 Algorithm 9: PC/M3/D

Similar to Algorithms 7 and 8, Algorithm 9 can be split into two parts the first assigning c_k SKUs to the various picking lines and the second duplicating the d_k SKUs with the highest pick frequency on each new picking line k. Determing the order in which SKUs are assigned to a picking line is the same as for Algorithm 7 and 8, but SKUs are assigned to picking lines by means of assignment method 3 (RHLC assignment). As for Algorithms 7 and 8, the d_k SKUs on the new picking line k with the highest pick frequency (p_i) is duplicated ($l_i \leftarrow 2$ for the relevant SKUs) in order to fill the new picking line so that none of the available locations are left unused.

Algorithm 9: PC/M3/D

Input : The variables L, C and c_k and sets $\mathcal{S}, \mathcal{L}$ and $\mathcal{P}$.
Output : Sets $\mathcal{S}^*, \mathcal{L}^*$ and $\mathcal{P}^*$.

Sort the SKUs in $\mathcal{S}$ from highest to lowest pick frequency (p_i);
$j \leftarrow 0$;
while $j < n$ **do**
    for $k = 1$ *to* L **do**
        if $|\mathcal{L}_k|_\Sigma + l_{j+1} \leq C$ **then**
            $j = j + 1$;
            $\mathcal{S}_k \leftarrow \mathcal{S}_k \cup s_j \cup s_{n-j+1}$;
            $\mathcal{L}_k \leftarrow \mathcal{L}_k \cup l_j \cup l_{n-j+1}$;
            $\mathcal{P}_k \leftarrow \mathcal{P}_k \cup p_j \cup p_{n-j+1}$;
        end
    end
end
for $k = 1$ *to* L **do**
    if $C > |\mathcal{L}_k|_\Sigma$ **then**
        Sort SKUs in set $\mathcal{S}_k$ from highest pick frequency per location to lowest;
        $j \leftarrow 1$;
        while $C > |\mathcal{L}_k|_\Sigma$ **do**
            $l_j \leftarrow 2$;
            $j = j + 1$;
        end
    end
end
$\mathcal{S}^* \leftarrow \mathcal{S}_1 \cup \mathcal{S}_2 \cup \ldots \cup \mathcal{S}_L$;
$\mathcal{L}^* \leftarrow \mathcal{L}_1 \cup \mathcal{L}_2 \cup \ldots \cup \mathcal{L}_L$;
$\mathcal{P}^* \leftarrow \mathcal{P}_1 \cup \mathcal{P}_2 \cup \ldots \cup \mathcal{P}_L$;

6.5 Combined DBN list, cluster SKUs first, duplicate second

Petersen and Aase proved that batching (or clustering) has the largest impact on reducing total order fulfillment time [107], while Dukic and Oluic also concluded that the batching of orders has the greatest potential for reducing the travel distance in a warehouse [33]. Clustering is about discovering smaller groups within a large set of data and can be defined as: *"A partition of the data is sought, in which each individual or object belongs to a single cluster and the complete set of clusters contains all the individuals"* [42]. Based on these findings the effect of combining order batching and the duplication of SKUs on the picking lines is investigated in the remainder of this chapter by means of various order clustering algorithms.

All the clustering methods discussed follow a hierarchical clustering type flow where classification consists of a series of partitions, which may run from a single cluster to a total of n clusters if the combined DBN list contain n SKUs [42]. Hierarchical clustering mainly consists of two types, agglomerative clustering and divisive clustering. Agglomerative clustering methods start off with n clusters each containing a single individual and with each iteration clusters are joined together until a single cluster exists which contains all n individuals. Divisive clustering methods is the direct opposite of agglomerative clustering in the sense that it starts off with a single cluster containing all n individuals and with each iteration a single individual or a cluster of individuals are separated until n clusters remain each containing a single individual. Between the two hierarchical clustering methods mentioned, agglomerative clustering methods is the most widely used [42].

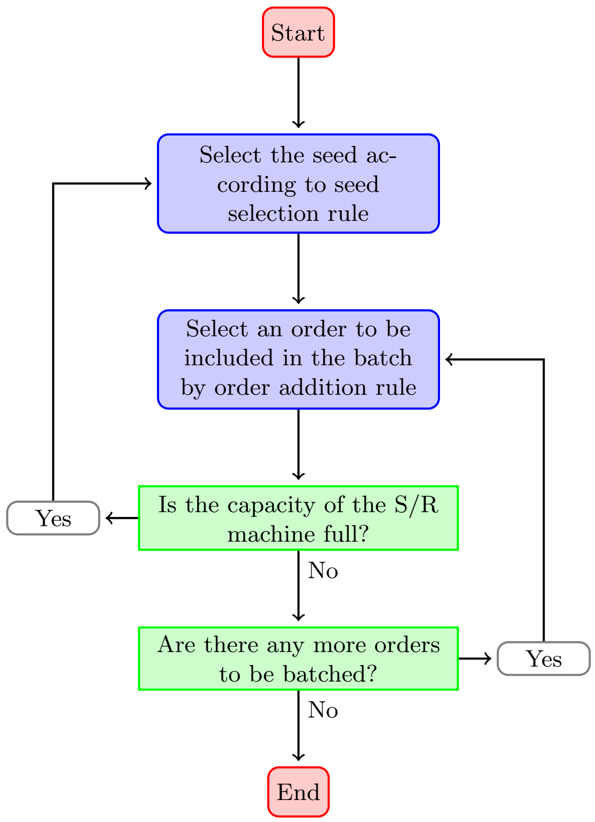


Figure 6.6: *An order batching algorithm procedure [98].*

For the algorithms in this section a typical order batching algorithm procedure is followed, as illustrated in Figure 6.6. For an order batching algorithm a set of orders need to be made up from products with a known storage location, and mainly consists of 2 steps, first to select a seed[1] and secondly the addition of subsequent orders to a batch [98]. The first two algorithms mainly cluster orders by branch, whereas the remaining algorithms in this section cluster by SKU. For all the algorithms in this section the SKUs are clustered in the picking lines until the picking line has reached its capacity and then filled by duplicating a sufficient number of SKUs with the highest pick frequency so that none of the locations are left empty.

As for Algorithms 3–9, the M original picking lines selected are combined into a single DBN list from which set $\mathcal{S}$ is constructed by listing all the unique SKUs in the combined DBN list. The decision flow for the algorithms discussed in this section is illustrated by Figure 6.7. The input files for these algorithms vary mainly due to the fact that some of the algorithms cluster SKUs in picking lines based on the branches while other cluster based on the SKU distribution. The number of SKUs to cluster in a picking line (c_k) for the algorithms discussed in this section, is determined by the number of picking lines to construct (L) and the number of SKUs on the original picking lines selected. Due to the fact that for Algorithms 12–15, the SKUs are clustered in picking lines two at a time, the number of SKUs that are to be clustered in the picking lines vary slightly of that of Algorithms 10 and 11. See Table 6.7 and Tables A.10–A.12.

The second part of the algorithms are exactly the same for all the algorithms discussed in §6.5.

[1]A seed is defined as the first order to be selected to form a batch [98].

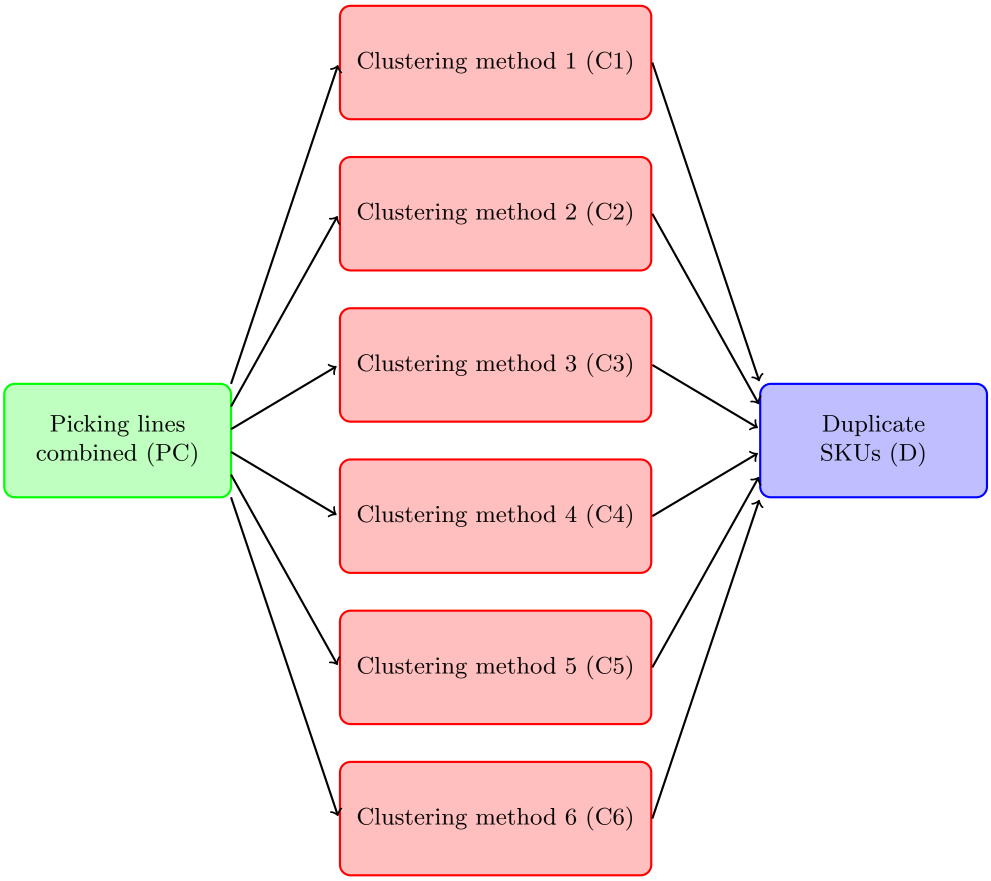


Figure 6.7: *A schematical representation of the decision flow when constructing Algorithms 10–15. Algorithms 10–15 combine the original picking lines selected and then SKUs are clustered in the picking lines by means of one of the clustering methods. Finally the SKUs on each of the new picking lines with the highest pick frequencies are duplicated.*

Nr picking lines (L)	Picking line (k)	# SKUs (c_k)	Duplications (d_k)
4	1	42	14
	2	42	14
	3	42	14
	4	42	14
5	1	34	22
	2	34	22
	3	34	22
	4	34	22
	5	32	24
6	1	28	28
	2	28	28
	3	28	28
	4	28	28
	5	28	28
	6	28	28

Table 6.7: *Number of SKUs to be clustered in each picking line and the number of duplications required to fill all the locations on each line, when selecting 3 of the original picking lines and applying Algorithms 12–15, if all the original picking lines contain 56 SKUs.*

Once the SKUs are clustered in the picking lines, the picking lines are filled by duplicating ($l_i \leftarrow 2$ for the relevant SKUs) the d_k SKUs with the highest pick frequency on each picking line so that all 56 locations on each of the picking lines are occupied, therefore $c_k + d_k = 56$, as for the algorithms discussed in §6.4. The number of SKUs to duplicate on picking line k is equal to

$$\begin{aligned} d_k &= C - c_k \\ &= C - |\mathcal{L}_k|_\Sigma, \end{aligned}$$

where $|\mathcal{L}_k|_\Sigma = \sum_{i=1}^{n} l_i$ is the total number of locations recuired to accommodate the SKUs clustered in picking line k. SKUs are thus duplicated until the number of locations required by the SKUs clustered in picking line k ($|\mathcal{L}_k|_\Sigma$) is equal to the capacity of the picking line ($C = 56$), thus when $d_k = 0$ and $C = |\mathcal{L}_k|_\Sigma$.

6.5.1 Algorithm 10: PC/C1/D

For Algorithm 10, instead of determining the pick frequency (or pick frequency per location) of each SKU on the DBN list, the number of SKUs to be distributed to each branch is counted (SKU density). This creates a clustering by branch effect or order batching. The reason for clustering by branch is to distribute branch orders over less picking lines, which increases the number of picks per cycle, resulting in less cycles in total.

The input file for Algorithm 10 is as for the algorithms in §6.3 with the addition of

- set $\mathcal{B} = \{b_1, b_2, \ldots, b_q, \ldots, b_\beta\}$ with b_q the branches to which the SKUs in set $\mathcal{S}$ is to be distributed to if they are to be distributed to a total of β branches,
- set $\mathcal{G} = \{g_1, g_2, \ldots, g_i, \ldots, g_n\}$ indicating whether a SKU i in set $\mathcal{S}$ is still to be clustered in a picking line, $g_i = 1$ if SKU i is unassigned and $g_i = 0$ if SKU i is assigned to a picking line
- the distribution matrix $\mathbf{F}$ where $f_{ib} = 1$ if SKU i is to be distributed to branch b and equal to 0 otherwise and

- the number of SKUs that are to be clustered in each picking line k is also indicated by c_k

Initially $g_i = 1$ for all SKUs in set $\mathcal{S}$ and when SKU i is clustered in a picking line then $g_i \leftarrow 0$. Therefore when all the SKUs in set $\mathcal{S}$ have been clustered in a picking line then $\sum_{i=1}^{n} g_i = 0$. Each $e_b \in \mathcal{E}$ is calculated as

$$e_b = \sum_{i=1}^{n} f_{ib}.$$

Thus set $\mathcal{E} = \{e_1, e_2, \ldots, e_q, \ldots, e_\beta\}$ is calculated with e_q indicating the total number of SKUs that are contained in set $\mathcal{S}$ is to be distributed to each of the branches contained in set $\mathcal{B}$ giving the SKU density of the branches.

As for the algorithms in §6.4, Algorithm 10 also consists of 2 parts, the first identifying the order in which the SKUs are to be clustered in the L picking lines and clustering them and the second duplicating the d_k SKUs with the highest pick frequencies on each picking line k ($l_j \leftarrow 2$ if SKU j is to be duplicated). For Algorithm 10 the seed is the branch for which e_b is a minimum. The branch with the minimum SKU density (e_b) that has not yet been assigned to a picking line is selected and not the maximum, due to the fact that some of the branches receive such a large number of SKUs that they can't all be accommodated on a single picking line which has only 56 loctions available ($C = 56$). Therefore set $\mathcal{B}$ is sorted in ascending order by the SKU density, e_b. If clustering the SKUs to be distributed to branch b ($f_{ib} = 1$) that has not yet been assigned to a picking line ($g_i = 1$) does not exceed the capacity of picking line k, in other words if $|\mathcal{L}_k|_\Sigma + \sum_{i=1}^{n} f_{ib} g_i \leq c_k$, then the relevant SKUs are assigned to picking line k and therefore all s_i, p_i and l_i for which $f_{ib} g_i \neq 0$ is removed from set $\mathcal{S}$, $\mathcal{P}$ and $\mathcal{L}$ and assigned to $\mathcal{S}_k$, $\mathcal{P}_k$ and $\mathcal{L}_k$ respectively. The subsequent branch to cluster to the seed branch, is the branch order with the next lowest SKU density. These SKUs are combined with the seed in order to enlarge the cluster. This is also referred to as the cumulative seeding rule [40], as the seed is updated to include the newly clustered SKUs. For Algorithm 10 the first picking line ($k = 1$) is filled to its capacity before starting to cluster SKUs to the next picking line for which the seed branch once again will be the remaining branch b for which e_b is a minimum. As branches are identified as the additional branch whose SKUs are to be clustered in the picking lines, they are removed from set $\mathcal{B}$ and assigned to set $\mathcal{B}^*$ to prevent them from being considered by the future iterations of the first part of the algorithm.

The second part of Algorithm 10 follows when c_k SKUs have been clustered in each of the picking lines k and all the SKUs in set $\mathcal{S}$ has been clustered each picking line k. In other words when $\sum_{i=1}^{n} g_i = 0$ and sets $\mathcal{S}$, $\mathcal{P}$ and $\mathcal{L}$ are all empty. The number of SKUs to be clustered in each of the picking lines is as for the algorithms in §6.4 where $c_k \approx \lceil \frac{n}{L} \rceil$. The second part considers each picking line k that was constructed in the first part and duplicate d_k SKUs in picking line k so that none of the available locations on the picking lines are left empty.

When there are multiple branches with the next lowest SKU density, all the branches are clustered in the new picking line if there is enough open locations available on the picking line. For instances where there is an insufficient number of locations available to accommodate more than one branch's SKUs, the branch that will fill the picking line to its capacity is selected, or the combination of branches that will fill the picking line to its capacity is selected.

Algorithm 10: PC/C1/D

Input : The variables L, C and c_k, sets $\mathcal{S}, \mathcal{L}, \mathcal{P}, \mathcal{B}$ and $\mathcal{G}$ and distribution matrix $\mathbf{F}$.
Output : Sets $\mathcal{S}^*, \mathcal{L}^*, \mathcal{P}^*$ and $\mathcal{B}^*$.

Sort set $\mathcal{B}$ in ascending order of SKU density (e_b);
$b \leftarrow 0$;
$k \leftarrow 1$;
while $b \leq q$ **do**
  while $|\mathcal{L}_k|_\Sigma + \sum_{i=1}^{n} f_{ib} g_i \leq c_k$ **do**
    for $i = 1$ *to* n **do**
      if $f_{ib} > 0$ **then**
        $\mathcal{S}_k \leftarrow \mathcal{S}_k \cup s_i$;
        $\mathcal{L}_k \leftarrow \mathcal{L}_k \cup l_i$;
        $\mathcal{P}_k \leftarrow \mathcal{P}_k \cup p_i$;
        $g_i \leftarrow 0$;
        $\mathcal{B}^* \leftarrow \mathcal{B} \cup b_b$;
      end
    end
    $b = b + 1$;
  end
  $k = k + 1$;
end
for $k = 1$ *to* L **do**
  if $C > |\mathcal{L}_k|_\Sigma$ **then**
    Sort SKUs in set $\mathcal{S}_k$ from highest pick frequency per location to lowest for picking line k;
    $j \leftarrow 1$;
    while $C > |\mathcal{L}_k|_\Sigma$ **do**
      $l_j \leftarrow 2$;
      $j = j + 1$;
    end
  end
end
$\mathcal{S}^* \leftarrow \mathcal{S}_1 \cup \mathcal{S}_2 \cup \ldots \cup \mathcal{S}_L$;
$\mathcal{L}^* \leftarrow \mathcal{L}_1 \cup \mathcal{L}_2 \cup \ldots \cup \mathcal{L}_L$;
$\mathcal{P}^* \leftarrow \mathcal{P}_1 \cup \mathcal{P}_2 \cup \ldots \cup \mathcal{P}_L$;

6.5.2 Algorithm 11: PC/C2/D

Algorithm 11 is a second branch clustering algorithm and follows the same decision flow as illustrated in Figure 6.6. The seed branch b is also selected as the branch with the lowest SKU density (e_b) as for Algorithm 10. The first part of Algorithm 11 also determines the order in which SKUs are clustered in the L picking lines while the second part duplicate the SKUs on each picking line k with the highest pick frequency in order to fill the picking lines to capacity as for Algorithm 10. The difference to Algorithm 10 comes in when selecting the additional branch whoose SKUs are to be clustered in the picking line after the seed branch. The order addition is then reiterated until the first picking line reaches its capacity (c_1). Then the next picking line is started by determing a new seed branch from the remaining branches in set $\mathcal{B}$.

The input file for Algorithm 11 is exactly the same as for Algorithm 10 as both algorithms cluster SKUs in picking lines by branch. The first part of Algorithm 11 therefore sort the branches in set $\mathcal{B}$ by ascending order of SKU density (e_b) to identify the order in which the branches' SKUs are to be clustered in the various picking lines. The seed branch is therefore the branch with the minimum SKU density as for Algorithm 10. All the SKUs that are to be distributed to the seed branch is then clustered in the first picking line. The minimum

SKU density branch is selected first as some of the branches is to receive such a large number of SKUs that they won't fit onto a single picking line and therefore additional clustering will need to be implemented in order to determine how the SKUs are to clustered in the picking lines.

The additional allocation rule differ from that of Algorithm 10 in the sense that Algorithm 10 clustered the SKUs that are to be distributed to the branch with the lowest SKU density (e_b) for which $g_i = 1$ to the picking lines next while for Algorithm 11 the number of additional SKUs ($\sum_{i=1}^{n} f_{ib} g_i$) that are to be clustered in the picking lines if branch b's SKUs that are clustered in the picking lines next are calculated and the branch with the minimum additional SKUs are clustered in the picking lines. The additional allocation rule is repeated until picking line k reaches its capacity c_k, after which the seed branch is identified for picking line $k+1$.

The final step is as for Algorithm 10, where the d_k SKUs with the highest pick frequency on picking line k is duplicated so that none of the locations is left unused. Tables 6.6 indicate the capacities ($c_k \approx \frac{n}{L}$) and number of duplications ($d_k = C - c_k$) for the various picking lines for instances when 3 of the original picking lines are selected and they all contain 56 SKUs (additional tables are provided in the appendix, Tables A.7 – A.9).

Algorithm 11: PC/C2/D

Input : The variables L, C and c_k, sets $\mathcal{S}, \mathcal{L}, \mathcal{P}, \mathcal{B}, \mathcal{E}$ and $\mathcal{G}$ and distribution matrix $\mathbf{F}$.
Output : Sets $\mathcal{S}^*, \mathcal{L}^*, \mathcal{P}^*$ and $\mathcal{B}^*$.

Sort set $\mathcal{B}$ in ascending order of SKU density (e_b);
$b \leftarrow 0$;
$k \leftarrow 1$;
while $b \leq q$ **do**
  Sort set $\mathcal{B}$ in ascending order of $\sum_{i=1}^{n} f_{ib} g_i$;
  while $|\mathcal{L}_k|_\Sigma + \sum_{i=1}^{n} f_{ib} g_i \leq c_k$ **do**
    for $i = 1$ *to* n **do**
      if $f_{ib} > 0$ **then**
        $\mathcal{S}_k \leftarrow \mathcal{S}_k \cup s_i$;
        $\mathcal{L}_k \leftarrow \mathcal{L}_k \cup l_i$;
        $\mathcal{P}_k \leftarrow \mathcal{P}_k \cup p_i$;
        $g_i \leftarrow 0$;
        $\mathcal{B}^* \leftarrow \mathcal{B} \cup b_b$;
      end
    end
    $b = b + 1$;
  end
  $k = k + 1$;
end
for $k = 1$ *to* L **do**
  if $C > |\mathcal{L}_k|_\Sigma$ **then**
    Sort SKUs in set $\mathcal{S}_k$ from highest pick frequency per location to lowest for picking line k;
    $j \leftarrow 1$;
    while $C > |\mathcal{L}_k|_\Sigma$ **do**
      $l_j \leftarrow 2$;
      $j = j + 1$;
    end
  end
end
$\mathcal{S}^* \leftarrow \mathcal{S}_1 \cup \mathcal{S}_2 \cup \ldots \cup \mathcal{S}_L$;
$\mathcal{L}^* \leftarrow \mathcal{L}_1 \cup \mathcal{L}_2 \cup \ldots \cup \mathcal{L}_L$;
$\mathcal{P}^* \leftarrow \mathcal{P}_1 \cup \mathcal{P}_2 \cup \ldots \cup \mathcal{P}_L$;

6.5.3 Algorithm 12: PC/C3/D

Algorithm 12 is another clustering algorithm where a similarity matrix $\mathbf{X}$ is constructed, so that x_{ij} indicate the number of branches to which both SKU i and j is to be distributed if there are n SKUs in set $\mathcal{S}$. The SKUs are clustered by assigning the 2 SKUs with the highest number of mutual branches to the picking lines first.

The input file for Algorithm 12 differs from the previous 2 algorithms as they cluster by branch and not SKU and is as for the algorithms discussed in §6.3 with the addition of

- the similarity matrix $\mathbf{X}$ with elements x_{ij} that is equal to the total number of branches to which both SKU i and j are to be distributed to, with $x_{ij} = x_{ji}$ and $x_{ij} = 0$ if $i = j$, and
- the number of SKUs that are to be clustered in each picking line k is also indicated by c_k

For Algorithm 12 the seed SKUs sre SKUs i and j for which x_{ij} is a maximum, in other words SKU i and j that are to be distributed to the highest mutual number of branches over all SKUs i and j. Once SKU i and j are clustered in a picking line they are removed from the similarity matrix $\mathbf{X}$, this ensures that SKUs are not taken into account for iterations after being clustered in a picking line and thus not assigned to more than one picking line. The additional allocation rule therefore is to cluster the remaining SKUs i and j for which x_{ij} is a maximum. All SKUs are clustered in picking line k until it reaches c_k.

As for the previous algorithms in this section, the second part of Algorithm 12, after all SKUs in set $\mathcal{S}$ have been clustered in a picking line, determines if picking line k has less SKUs assigned to it than required to fill all the available locations. If the SKUs clustered in picking line k require less than 56 locations the SKU with the highest pick frequency per location ($\frac{p_{kj}}{l_{kj}}$) is duplipcated ($l_{kj} \leftarrow 2$). The SKUs with the highest pick frequency per location is duplicated untill the number of locations required by the SKUs clustered in picking line k ($|\mathcal{L}_k|_\Sigma$) is equal to the capacity of the picking line.

6.5.4 Algorithm 13: PC/C4/D

In some instances it is better to determine the number of branches for which two SKUs are equally distributed to rather than only the number of branches to which both are to be distributed to and therefore excluding the cases when both SKUs are not to be distributed to a branch [42], as in Algorithm 12. Algorithm 13 is therefore also a clustering algorithm where a dissimilarity matrix $\mathbf{Y}'$ is constructed, such that y'_{ij} indicate the number of branches to which either SKU i or j is distributed to.

The input file for Algorithm 13 is as for Algorithm 12 except for the simimlarity matrix is replaced by a dissimilarity matrix $\mathbf{Y}'$ with elements y'_{ij} that is equal to the number of branches to which either SKU i or j is to be distributed to and $y'_{ij} = y'_{ji}$ while $y'_{ij} = b$ if $i = j$ (where b is the total number of branches in the combined DBN list to which the SKUs in set $\mathcal{S}$ is to be distributed to). The inclusive similarity matrix $\mathbf{Y}$ differs from the similarity matrix $\mathbf{X}$ in Algorithm 12 in the sense that y_{ij} indicates the number of branches to which both SKUs i and j are distributed and not distributed to, whereas x_{ij} only indicates the number of branches to which both SKU i and j are distributed to. Therefore x_{ij} will always be less than or equal to

Algorithm 12: PC/C3/D

Input : The variables L, C and c_k, sets $\mathcal{S}, \mathcal{L}$ and $\mathcal{P}$ and similarity matrix $\mathbf{X}$.
Output : Sets $\mathcal{S}^*, \mathcal{L}^*$ and $\mathcal{P}^*$.

$i \leftarrow 1$;
$j \leftarrow i + 1$;
$k \leftarrow 1$;
while $|\mathcal{L}_k|_\Sigma + l_{j+1} \leq c_k$ **do**
    for $i = 1$ *to* n **do**
        for $j = i + 1$ *to* n **do**
            if x_{ij} *is the maximum* **then**
                $\mathcal{S}_k \leftarrow \mathcal{S}_k \cup s_i \cup s_j$;
                $\mathcal{L}_k \leftarrow \mathcal{L}_k \cup l_i \cup l_j$;
                $\mathcal{P}_k \leftarrow \mathcal{P}_k \cup p_i \cup p_j$;
                Remove SKUs i and j from similarity matrix $\mathbf{X}$;
            end
        end
    end
    $k = k + 1$;
end
for $k = 1$ *to* L **do**
    if $C > c_k$ **then**
        Sort SKUs in set $\mathcal{S}_k$ from highest pick frequency per location to lowest for picking line k;
        $j \leftarrow 1$;
        while $C > |\mathcal{L}_k|_\Sigma$ **do**
            $l_{kj} \leftarrow 2$;
            $j = j + 1$;
        end
    end
end
$\mathcal{S}^* \leftarrow \mathcal{S}_1 \cup \mathcal{S}_2 \cup \ldots \cup \mathcal{S}_L$;
$\mathcal{L}^* \leftarrow \mathcal{L}_1 \cup \mathcal{L}_2 \cup \ldots \cup \mathcal{L}_L$;
$\mathcal{P}^* \leftarrow \mathcal{P}_1 \cup \mathcal{P}_2 \cup \ldots \cup \mathcal{P}_L$;

y_{ij} and $y_{ij} = 0$ if $i = j$.

$$\mathbf{Y} = \begin{bmatrix} y_{11} & y_{12} & \dots & y_{1n} \\ y_{21} & y_{22} & \dots & y_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ y_{n1} & y_{n2} & \dots & y_{nn} \end{bmatrix} = \begin{bmatrix} 0 & y_{12} & \dots & y_{1n} \\ y_{21} & 0 & \dots & y_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ y_{n1} & y_{n2} & \dots & 0 \end{bmatrix} \geq \mathbf{X}$$

The dissimilarity matrix $\mathbf{Y}'$ will therefore indicate the difference between the number of branches on the combined DBN list (b), which is also the maximum number of branches to which SKUs i and j can be distributed, and the number of branches to which both SKUs i and j are distributed and not distributed to, thus $y'_{ij} = b - y_{ij}$.

$$\begin{aligned} \mathbf{Y}' = \begin{bmatrix} y'_{11} & y'_{12} & \dots & y'_{1n} \\ y'_{21} & y'_{22} & \dots & y'_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ y'_{n1} & y'_{n2} & \dots & y'_{nn} \end{bmatrix} &= \begin{bmatrix} b - y_{11} & b - y_{12} & \dots & b - y_{1p} \\ b - y_{21} & b - y_{22} & \dots & b - y_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ b - y_{p1} & b - y_{p2} & \dots & b - y_{pp} \end{bmatrix} \\ &= \begin{bmatrix} b & b - y_{12} & \dots & b - y_{1n} \\ b - y_{21} & b & \dots & b - y_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b - y_{n1} & b - y_{n2} & \dots & b \end{bmatrix} \end{aligned}$$

For Algorithm 13 the seed SKUs are SKUs i and j for which y'_{ij} is a minimum, in other words SKU i and j that are distributed to the least number of branches that will only recieve SKU i or SKU j. Once SKU i and j are clustered in a picking line they are removed from the dissimilarity matrix $\mathbf{Y}'$, this ensures that SKUs are not taken into account for iterations after being clustered in picking lines and thus not assigned to more than one picking line. The additional allocation rule therefore is to cluster the remaining SKUs i and j for which y'_{ij} is a minimum. The second part of Algorithm 13 is exaclty the same as for Algorithm 12.

Algorithm 13: PC/C4/D

Input : The variables L, C and c_k, sets $\mathcal{S}, \mathcal{L}$ and $\mathcal{P}$ and dissimilarity matrix $\mathbf{Y}'$.
Output : Sets $\mathcal{S}^*, \mathcal{L}^*$ and $\mathcal{P}^*$.

```
i ← 1;
j ← i + 1;
k ← 1;
while |L_k|_Σ + l_{j+1} ≤ c_k do
    for i = 1 to n do
        for j = i + 1 to n do
            if y'_{ij} is the minimum then
                S_k ← S_k ∪ s_i ∪ s_j;
                L_k ← L_k ∪ l_i ∪ l_j;
                P_k ← P_k ∪ p_i ∪ p_j;
                Remove SKUs i and j from dissimilarity matrix Y';
            end
        end
    end
    k = k + 1;
end
for k = 1 to L do
    if C > c_k then
        Sort SKUs in set S_k from highest pick frequency per location to lowest for picking line k;
        j ← 1;
        while C > |L_k|_Σ do
            l_{kj} ← 2;
            j = j + 1;
        end
    end
end
S* ← S_1 ∪ S_2 ∪ ... ∪ S_L;
L* ← L_1 ∪ L_2 ∪ ... ∪ L_L;
P* ← P_1 ∪ P_2 ∪ ... ∪ P_L;
```

6.5.5 Algorithm 14: PC/C5/D (Alternative to Algorithm 12)

Algorithm 14 can be seen as an alternative to Algorithm 12, in the sense that it only differs when it comes to the order in which the picking lines are filled. Algorithm 12 filled the first picking line to c_k SKUs, before starting to cluster SKUs to the second picking line and so forth until all the SKUs in set $\mathcal{S}$ have been clustered in a picking line and all picking lines have been filled to its capacity. Whereas for Algorithm 14, the SKUs are clustered in the picking lines by cyclical assignment as for Algorithm 4 and 7 described. The pseudo code for this algorithm is given in Appendix B.

6.5.6 Algorithm 15: PC/C6/D (Alternative to Algorithm 13)

Simmilarly Algorithm 15 can be seen as an alternative to Algorithm 13, in the sense that it only differs when it comes to the order in which the picking lines are filled. Algorithm 13 filled the first picking line to c_k SKUs. Whereas for Algorithm 15, the SKUs are clustered in picking lines by cyclical assignment as for Algorithm 4 and 7. The pseudo code for this algorithm is given in Appendix B.

6.6 Conclusion

In this chapter 15 algorithms are discussed based on different clustering or batching methods. In §6.2 the original picking lines received from PEP is used as a starting point for the two algorithms. Due to the fact that PEP assign SKUs from a combined DBN list and the fact that it provides more opportunity to optimise, a combined DBN list is used in the remainder of the chapter.

While the algorithms in §6.3 and §6.4 assign the SKUs to the picking lines in the same manner with the only difference being that in §6.3 the SKUs with the highest pick frequencies are duplicated before assigning them to the picking lines while the algorithms in §6.4 assign the SKUs to the various picking lines and only then duplicated the SKUs in each picking line with the highest pick frequencies in order to fill all the available locations within each picking line. This is to determine whether it is more beneficial in terms of picker travel distance to first duplicate the SKUs on the combined DBN list and then assign SKUs to the picking lines or assign the SKUs to the picking lines first and then duplicate the SKUs with the highest pick frequency on each picking line.

Algorithms 3 and 4 only differ in the sense that the non-duplicate and duplicate SKUs are assigned to the various picking lines seperately for Algorithm 3 while Algorithm 4 first sort the SKUs by pick frequency per location before assigning them simultaneously. This is in order to see the effect of treating duplicated and non-duplicated SKUs seperately on the travel distance of the pickers within the order picking system. The three methods for assigning the SKUs to the various picking lines that are investigated is by means of cyclical assignment (M1), set length subset sequential assignment (SLSS, M2) and remaining high, low cyclical assignment (RHLC, M3). These three methods are implemented by the algorithms in §6.3 and §6.4.

Algorothms 10–15 first identify either a seed branch or SKUs to assign to a picking line and then cluster SKUs to the picking line based on an additional allocation rule. Algorithms 10 and 11 identify a seed branch with the minimum SKU density and cluster all the SKUs required by that branch to a picking line. The additional allocation rule for Algorithm 10 determines the branch with the minimum SKU density excluding all branches previously identified. For Algorithm 11 the additional allocation rule determines the branch that will assign the least number of additional SKUs to the picking lines. Algorithm 12 uses a similarity matrix and cluster SKUs in the order of the maximum mutual branches while Algorithm 13 makes use of a dissimilarity matrix and cluster the SKUs in the order of minimum dissimilar branch distribution. All four these algorithms cluster SKUs to a picking line until it reaches its capacity before assigning to the next picking line, SLSS. Algorithms 14 and 15 are variations of Algorithms 12 and 13 respectively. Where algorithms 12 and 13 cluster SKUs in picking lines by filling a picking line to its capacity before assigning SKUs to the next picking line, for Algorithms 14 and 15

SKUs are clustered in picking lines in the same order but by means of cyclical assignment as for Algorithms 3, 4 and 7.

CHAPTER 7

Results

Contents

In this chapter the results of the algorithms introduced in Chapter 6 are discussed. The picking lines constructed by the various algorithms were run through the Picking Line Solver (programmed by Matthews [90]) as for the experimental picking lines in Chapter 5 to determine what the minimum number of cycles is to complete the order picking for each constructed picking line. The algorithms was coded in JAVA by means of Netbeans IDE 6.8 [96] to construct the various picking lines and was run on a Pentium (R) Dual-Core 2.10GHz. The runtime for the majority of the instances was less than 15 seconds. For the clustering algorithms the runtime varied from 5 minutes to 10 minutes. Because the DC has around 30 minutes available to determine good solutions, all these runtimes are acceptable. Therefore the algorithms are not compared by means of solution times, but rather by means of solution quantity (i.e. the number of cycles).

It is sufficient to compare results only based on the number of cycles it would take the pickers to complete the order picking on all the picking lines as a measure of solution quality. This is mainly because the number of products are constant within a group which means that there is no increase in the amount of time it takes to construct the number of picking lines as the same number of products are removed from the storage racks and only placed in more locations with an increase in the number of duplications. Therefore a saving in the number of cycles will decrease the time it takes to complete the order picking and is the total saving achieved.

7.1 Input data

As previously mentioned, PEP supplied 22 original picking lines that was implemented at the Durban DC. These were catagorised as small, medium and large picking lines based on their pick frequency distribution and number of SKUs on the picking line as indicated in Table 5.1. The original picking lines A–J and L (the 11 largest original picking lines received from PEP)

Group	#Original picking lines	Original picking lines	Minimum cycles for group	Total #SKUs	Average cycles per picking line
A-5	5	B, D, F, J, L	4302	261	860.4
B-3	3	D, J, L	2194	153	731.3
C-3	3	C, H, I	3078	161	1026.0
D-3	3	E, G, H	3043	158	1014.3
E-3	3	A, B, F	3340	152	1113.3
F-3	3	A, E, I	3253	156	1084.3
G-3	3	F, G, J	2830	164	943.3
H-3	3	B, C, D	3409	161	1136.3
I-4	4	A, B, C, F	4501	200	1125.3
J-4	4	D, H, I, J	3880	222	970.0
K-4	4	A, E, G, I	4260	209	1065.0
L-4	4	C, D, F, J	4006	213	1001.5
M-4	4	B, C, E, J	4398	209	1099.5
N-4	4	D, F, G, J	3852	220	963.0
O-4	4	B, E, H, I	4213	214	1053.3
P-2	2	A, F	2114	104	1057.0
Q-2	2	H, I	1917	110	958.5
R-2	2	B, G	2233	107	1116.5
S-2	2	C, J	2102	105	1051.0
T-2	2	D, E	2092	107	1046.0
U-2	2	A, H	2198	103	1099.0
V-2	2	B, J	2167	110	1083.5

Table 7.1: *Groups A-5 to V-2 is formed by selecting the original picking lines as specified, as well as the minimum number of cycles it would take the pickers to complete the order picking as per the Picking Line Solver and the number of unique SKUs contained in each group.*

was used to form groups of picking lines that was used to investigate the effects of implementing the algorithms in Chapter 6 on the number of cycles that it would take the pickers to complete the picking. The groups formed by randomly selecting between 2 and 5 of the original picking lines is given in Table 7.1. The algorithms in Chapter 6 was implemented for each of the groups and for an increase of 0–100% in the number of locations (it is 0–100% duplications).

7.2 Total number of cycles traversed

The results obtained from the Picking Line Solver indicated the number of cycles it would take to complete the order picking for each picking line when implementing each of the algorithms in Chapter 6. Figures C.1–C.15 in the appendix illustrate the total number of cycles it would require to complete the order picking for groups A-5 to V-2 with an increase in the number of picking lines for Algorithms 1–15 respectively. A summary of the percentage saving in the number of cycles and the percentage increase in the number of locations, for all groups and algorithms is given in Table 7.2. The percentage increase in the number of locations is equal to the percentage SKUs that is duplicated. All the results discussed in this chapter are based upon these results.

Table 7.2 present the maximum percentage saving in the number of cycles that is possible for all the groups when implementing the various algorithms and increasing the number of picking lines. The percentage increase in the number of locations (which is equal to the percentage SKUs duplicated) at which the minimum cycles is achieved for each group and algorithm is also given. The average percentage saving in the number of cycles, the average percentage SKUs duplicated and the average percentage increase in the picking lines are summarised in Table 7.3.

Figure 7.1 illustrate that the average percentage saving in the minimum number of cycles is mostly higher for the larger groups and that the smaller groups has a smaller percentage saving. It is also illustrated that Algorithms 5, 8 and 12 yields the greatest average saving. From Figure 7.2 it is visible that these 3 algorithms yields the greatest average percentage saving in the number of cycles. Algorithms 1, 2 and 10 gives the next best grouping of average percentage savings in the number of cycles. While the average percentage saving in the number of cycles for the remainder of the Algorithms are more or less the same. The algorithms may thus be divided into three groups, Group A1 containing Algorithms 5, 8 and 12 for which a high percentage saving in the number of cycles is achieved, Group A2 containing Algorithms 1, 2 and 10 for which a medium percentage saving in the number of cycles is achieved, and Group A3 containing the remainder of the algorithms for which a low percentage saving in the number of cycles is achieved.

From Table 7.2 it is clear that the various algorithms yield different percentage savings in the number of cycles for each groups and that the minimum number of cycles is also achieved at different numbers of picking lines. Overall for 18 of the 22 groups, Algorithm 5 yields the biggest saving in the number of cycles (see Table 7.4), while Algorithm 8 yield the minimum number of cycles for 11 groups and Algorithm 12 yield the minimum number of cycles for 1 group (there are groups where more than 1 algorithm yielded the minimum number of cycles). For 4 of the 22 groups the next best saving in the number of cycles was achieved by implementing Algorithm 5, while Algorithm 8 yielded the next best saving in the number of cycles for 7 groups. Algorithm 12 yielded the next best saving for 2 groups and Algorithm 10 yielded the next best saving for 1 group. The third best saving in the number of cycles was achieved for 4 groups by implementing Algortihm 8, while Algorithm 12 yielded the third best saving for 18 groups and Algorithm 14 yielded the third best saving in the number of cycles for the remaining group. Therefore it is concluded that Algorithm 5 outperforms the other algorithms in terms of the minimum total number of cycles, the majority of the time while Algorithm 8 follow closely. Algorithm 12 performs third best in terms of the total minimum number of cycles while the remainder of the algorithms mostly yield the fourth least minimum number of cycles. When excluding Algorithms 5 and 8 from Table 7.4, Algorithm 12 achieve the minimum number of cycles for 20 groups, as in Table 7.5. It is also visible that the clustering algorithms (Algorithms 10–15) outperforms the remainder of the algorithms. Table 7.6 represent the results if the clustering algorithms are excluded as well. The algorithms implementing PS (Algorithms 1 and 2) now yields the minimum number of cycles for the majority of the groups.

7.3 Percentage SKUs duplicated

From Figures C.1–C.15 and Table 7.2 it is clear that the minimum number of cycles for each group is not achieved by constructing the same number of picking lines for all the algorithms. There are similarities in the number of SKUs duplicated for the algorithms in Groups A1 to A3 that is visible in Figure 7.3.

Figure 7.4 present the cumulative number of groups for which the minimum number of cycles is achieved by a percentage increase in the number of picking lines. From Figure 7.4 it is visible that the results of the various algorithms can be one of 2 scenarios in terms of the increase in the number of picking lines that yield the minimum number of cycles for the various

	At minimum # cycles	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
1	% Saving in cycles	34	30	29	30	23	13	32	34	14	31	26	6	28	36	24	33	12	30	22	20	25	18
	% Increase in locations	62	96	74	77	42	44	71	74	34	77	61	28	59	78	58	62	55	57	59	59	63	55
2	% Saving in cycles	27	10	22	27	31	26	37	30	32	26	27	26	23	37	25	34	22	34	25	28	23	27
	% Increase in locations	62	67	74	75	75	77	67	99	88	100	88	54	85	27	100	62	100	100	100	100	63	98
3	% Saving in cycles	21	10	15	22	23	13	20	22	23	23	17	18	20	26	25	13	13	24	17	15	16	19
	% Increase in locations	21	33	74	77	42	44	71	39	34	51	34	54	58	53	56	62	100	57	55	57	63	53
4	% Saving in cycles	21	9	15	22	23	11	22	19	20	26	14	18	20	26	28	13	12	25	20	19	15	17
	% Increase in locations	21	33	74	77	42	44	71	39	34	51	34	54	58	53	56	62	100	57	55	57	63	53
5	% Saving in cycles	42	41	37	40	46	41	46	44	46	41	45	48	46	44	41	40	30	41	36	35	31	35
	% Increase in locations	62	67	74	100	77	79	100	74	88	100	88	56	85	78	82	100	100	100	96	100	100	100
6	% Saving in cycles	21	9	16	21	23	12	21	19	22	27	17	18	20	26	27	13	12	24	18	16	16	20
	% Increase in locations	21	33	74	77	42	44	37	39	34	51	34	28	58	53	56	62	100	57	53	57	63	53
7	% Saving in cycles	22	10	16	22	24	14	22	20	21	22	18	18	20	26	25	12	11	26	17	16	15	17
	% Increase in locations	21	33	74	77	42	44	71	39	34	51	34	54	32	53	56	62	100	57	57	57	63	53
8	% Saving in cycles	45	39	37	39	36	38	45	45	41	41	42	45	41	42	40	40	31	41	36	37	31	35
	% Increase in locations	88	82	100	100	77	100	100	100	88	100	100	95	97	100	99	100	100	100	96	100	100	100
9	% Saving in cycles	22	10	16	21	22	16	8	21	21	25	19	20	20	26	24	13	13	25	15	17	15	16
	% Increase in locations	21	33	74	77	42	44	100	39	34	51	34	28	58	53	56	62	100	57	57	57	63	53
10	% Saving in cycles	34	28	25	28	38	32	31	34	36	26	30	32	24	32	26	27	13	22	23	21	20	24
	% Increase in locations	62	67	74	77	77	79	71	74	88	51	88	80	85	78	82	100	100	100	57	57	63	100
11	% Saving in cycles	20	11	17	25	22	9	22	22	22	26	17	21	26	25	28	13	12	23	19	20	16	16
	% Increase in locations	42	33	74	77	42	44	37	39	34	51	34	54	58	53	82	62	100	57	57	57	63	53
12	% Saving in cycles	35	37	36	38	36	37	42	38	38	37	44	40	42	44	39	37	29	37	30	32	29	31
	% Increase in locations	62	82	100	100	77	100	100	74	61	100	88	95	85	100	99	100	100	100	96	100	100	100
13	% Saving in cycles	20	6	12	23	26	19	24	16	21	20	18	19	28	18	24	16	20	17	14	16	4	27
	% Increase in locations	42	67	100	77	77	79	100	100	88	100	61	54	85	53	82	62	100	57	96	100	100	100
14	% Saving in cycles	41	7	26	25	24	14	26	24	23	25	19	24	28	26	29	13	13	25	25	18	15	19
	% Increase in locations	82	33	74	77	42	44	71	74	34	51	61	28	58	53	56	62	100	57	57	57	63	53
15	% Saving in cycles	22	17	24	24	26	16	23	25	28	26	19	29	25	27	29	20	14	25	23	19	18	20
	% Increase in locations	42	33	74	77	42	44	71	74	61	51	34	54	85	53	56	61	100	57	57	57	63	100

Table 7.2: *Percentage saving in cycles and percentage increase in the number of locations for each group A–V and algorithm (listed in the first column) at which the minimum number of cycles is achieved.*

Algorithm	Decision flow	Saving in cycles	Average percentage Increase in picking lines	Duplications
1	PS/D/M1-ND	25.0	54.1	61.1
2	PS/D/M1-S	27.2	78.0	80.0
3	PC/D/M1-ND	18.9	47.7	54.0
4	PC/D/M1-S	18.9	47.7	54.0
5	PC/D/M2-S	40.7	83.6	86.6
6	PC/D/M3-S	19.0	45.1	51.2
7	PC/M1/D	18.8	46.6	52.9
8	PC/M2/D	39.4	97.3	96.5
9	PC/M3/D	18.4	46.6	54.2
10	PC/C1/D	27.5	71.5	77.7
11	PC/C2/D	19.7	48.3	54.7
12	PC/C3/D	36.7	90.5	91.8
13	PC/C4/D	18.5	76.3	80.9
14	PC/C5/D	22.2	52.0	58.5
15	PC/C6/D	22.7	54.7	61.2

Table 7.3: *Categorising the algorithms based on their average percentage saving in the number of cycles, increase in number of picking lines and increase in the number of locations.*

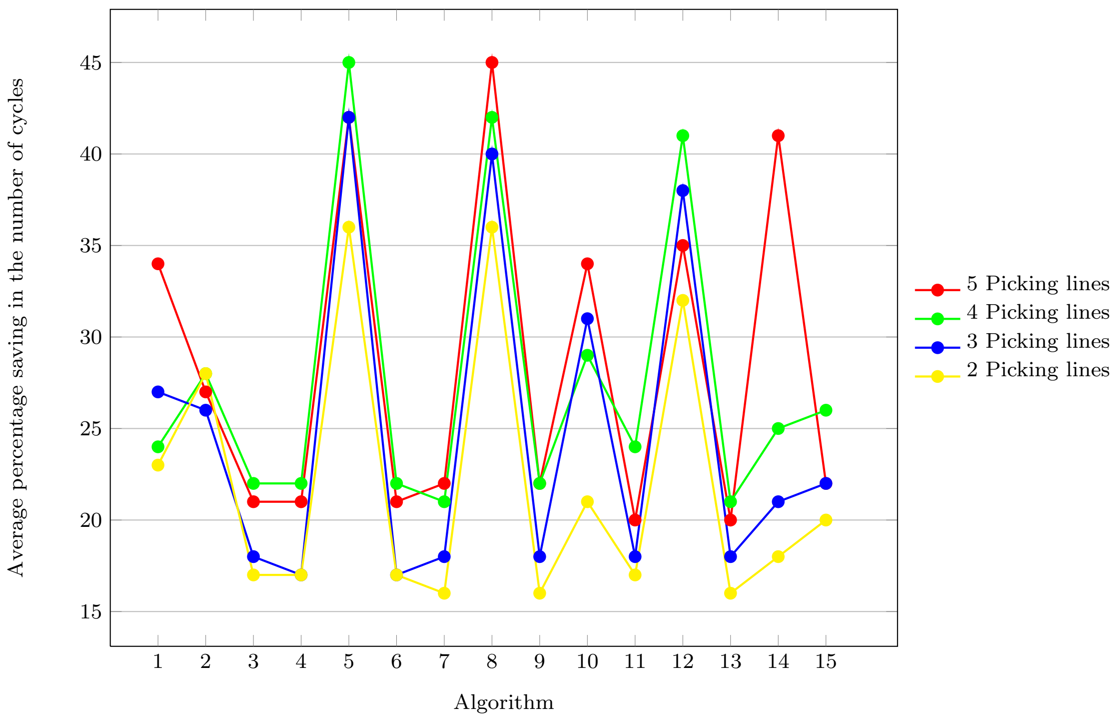


Figure 7.1: *The average percentage saving in the minimum number of cycles required for the groups consisting of 2, 3, 4 and 5 picking lines, when implementing the various algorithms presented in Chapter 6 as in Table 7.2 and Figure C.16.*

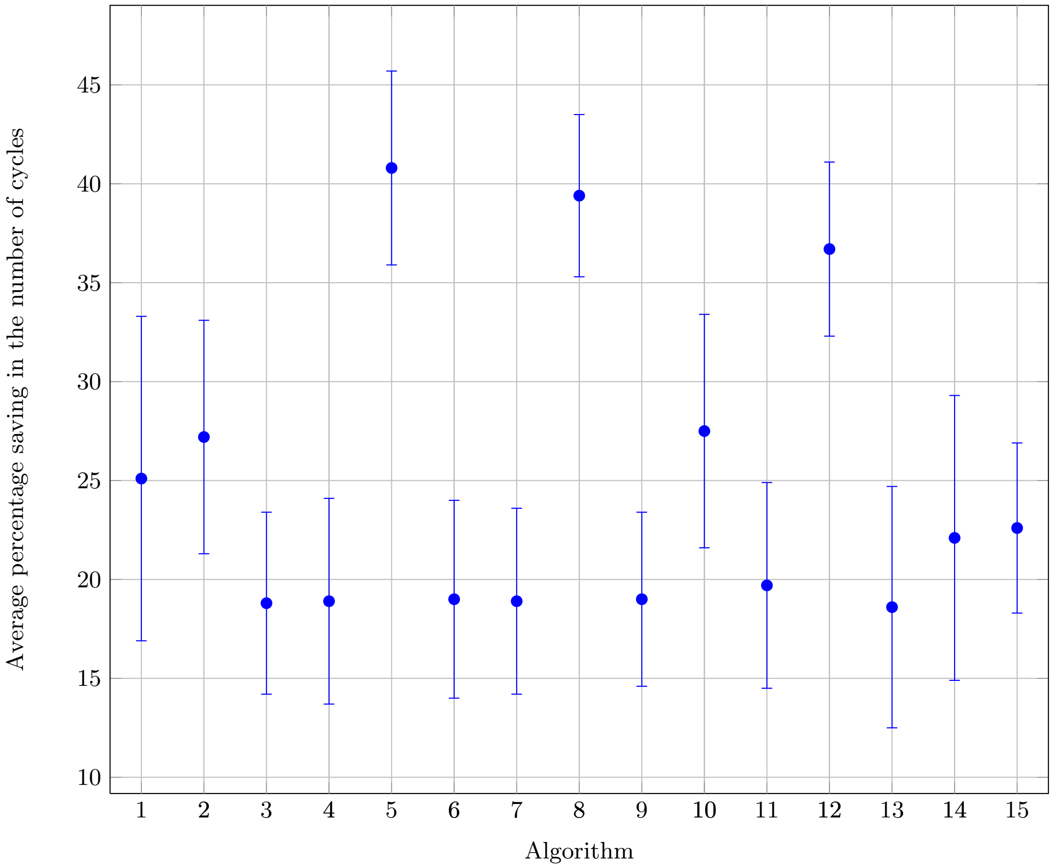


Figure 7.2: *The average percentage saving in the minimum number of cycles when implementing the various algorithms presented in Chapter 6 as in Table 7.2 and the standard deviation in the percentage saving.*

Algorithm	Minimum cycles	2nd Least cycles	3rd Least cycles	Other
1	–	–	–	22
2	–	–	–	22
3	–	–	–	22
4	–	–	–	22
5	18	4	–	–
6	–	–	–	22
7	–	–	–	22
8	11	7	4	–
9	–	–	–	22
10	–	1	–	21
11	–	–	–	22
12	1	2	18	–
13	–	–	–	22
14	–	–	1	21
15	–	–	–	22

Table 7.4: *The number of groups for which the minimum, second least and third least total number of cycles required to complete the order picking, is achieved by the various algorithms presented in Chapter 6.*

Algorithm	Minimum cycles	2nd Least cycles	3rd Least cycles	Other
1	–	7	5	10
2	–	8	5	9
3	–	–	–	22
4	–	–	–	22
6	–	–	1	21
7	–	–	–	22
9	–	–	–	22
10	1	5	4	12
11	–	–	–	22
12	20	2	–	–
13	–	2	1	19
14	1	3	1	17
15	–	1	1	20

Table 7.5: *The number of groups for which the minimum, second least and third least total number of cycles required to complete the order picking, is achieved by the various algorithms (excluding algorithms implementing M2) presented in Chapter 6.*

Algorithm	Minimum cycles	2nd Least cycles	3rd Least cycles	Other
1	8	7	1	6
2	13	7	2	–
3	–	3	10	9
4	1	–	10	11
6	–	3	8	11
7	–	2	10	10
9	–	4	7	11

Table 7.6: *The number of groups for which the minimum, second least and third least total number of cycles required to complete the order picking, is achieved by the various algorithms (excluding algorithms implementing M2 or clustering) presented in Chapter 6.*

groups. The first is that the biggest saving in the number of cycles is achieved with a small increase in the number of picking lines, a small percentage of the SKUs are duplicated. Between 14 and 19 of the 22 groups investigated in this study achieved their minimum number of cycles with an increase of less than or equal to 50% in the number of picking lines for these algorithms. The algorithms for which this is true are Algoritms 1, 3, 4, 6, 7, 9, 11, 14 and 15 (Group A3 from §7.2). The second scenario is that the biggest saving in the number of cycles is achieved by duplicating a significant number of SKUs, 60% or more SKUs are duplicated. Between 16 and 22 of the groups' minimum number of cycles is achieved by an increase of more than 60% in the number of picking lines. The algorithms for which this is true are Algorithms 2, 5, 8, 10, 12 and 13 (Groups A1 and A2 from §7.2). In Figure 7.5 the average percentage SKUs duplicated are presented as well as the standard deviation for each of the algorithms at the minimum number of cycles. This implies together with Figure 7.5 that Groups A1 and A2 require (on average) that a higher percentage of SKUs are duplicated than for Group A3. Thus the bigger the saving in the number of cycles the higher the number of SKUs that are duplicated.

From Figure 7.3 it is visible that the small groups of DBNs (consisting of 2 of the original picking lines), require that a bigger percentage of SKUs are duplicated than for the larger groups of DBNs (consisting of 5 of the original picking lines) for the majority of the picking lines. Figures 7.6–7.8 illustrate the average number of cycles with an increase in the number of picking lines for Algorithms 5, 2 and 7 respectively, as representatives for Groups A1 to A3 as the algorithms for each of the groups follow the same trend. It is visible from Figures 7.6–7.8 that Group A1 requires a bigger increase in the number of picking lines to achieve the average minimum cycles while Group A3 require only an increase of a few picking lines, while Group A2 require an increase between that of Groups A1 and A3 in the number of picking lines. There are thus similarities between the algorithms in terms of number of cycles and increase in the number of picking lines for Groups A1 to A3 and the decision flow implemented to construct the various algorithms in each of the groups as discussed in §6.1. Therefore it is necessary to compare the algorithms in terms of the number of cycles based on the decision flow implemented.

Figures 7.9–7.11 represent the maximum saving achieved by the various algorithms for the specified groups and the percentage of SKUs that is duplicated in order to achieve the maximum savings. It is visible that the greater the number of SKUs that is duplicated the greater the saving in the number of cycles. For some of the groups there is very little or no savings achieved with duplicating a small number of SKUs while for duplicating all the SKUs in the groups there is a decrease in the savings for some groups. Thus it may be concluded that by duplicating approximately 40–70% of the SKUs in a group a significant saving in the number of cycles will be achieved.

7.4 Impact of the decision flow

From Table 7.3, Figure 7.1 and 7.5 it is clear that there are some similarities in the percentage saving in cycles and percentage SKUs duplicated and percentage increase in picking lines if there are similarities in the decision flow of the algorithms. Therefore it is necessary to compare algorithms who share certain decision flow similarities in more detail based on their maximum percentage saving in the number of cycles and the percentage SKUs duplicated at which the minimum number of cycles is achieved for the various groups.

Figure 7.12 give the average percentage saving in the minimum number of cycles for the al-

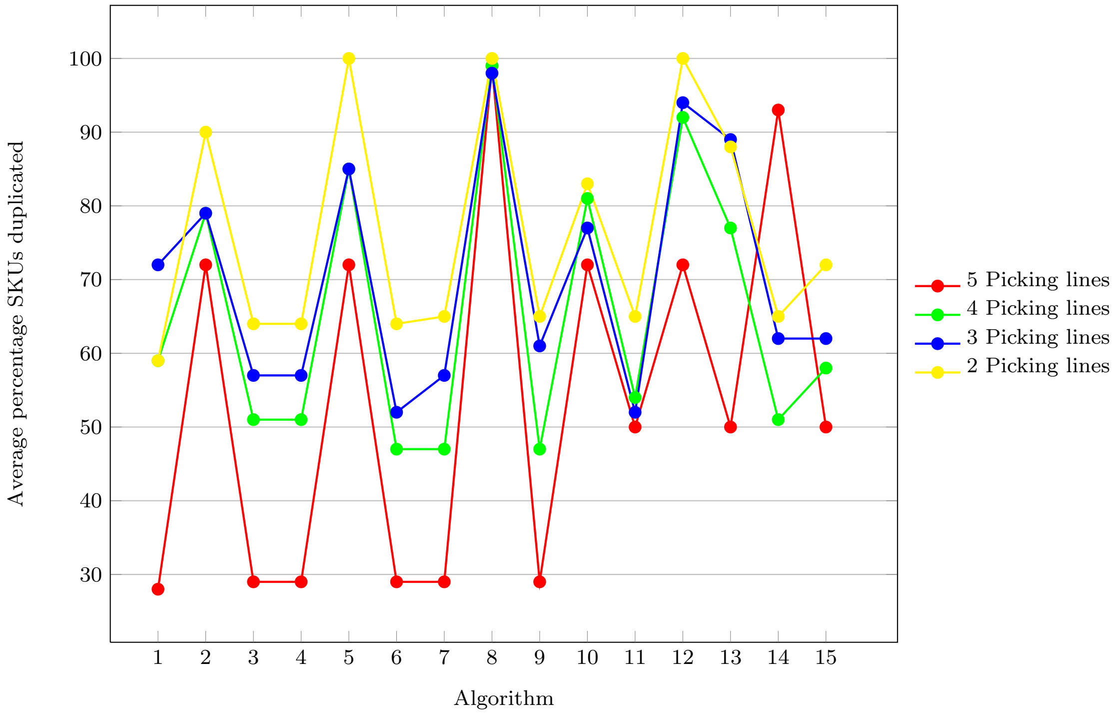


Figure 7.3: *Average percentage SKUs duplicated that yield the minimum number of cycles when implementing the various algorithms presented in Chapter 6 as in Table 7.2.*

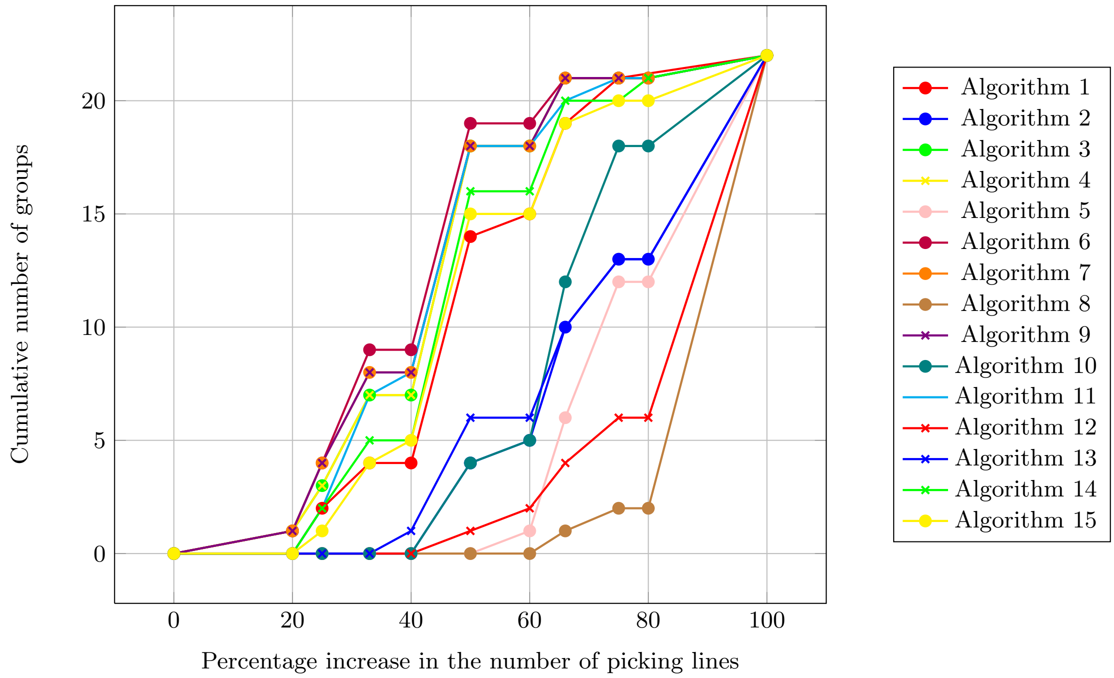


Figure 7.4: *A graph representing the cumulative number of groups and the corresponding percentage increase in the number of picking lines yielding the minimum number of cycles for all the algorithms discussed in Chapter 6. There is either a significant saving in the number of cycles with small increase in the number of picking lines or a significant saving in the number of cycles with a large increase in the number of picking lines when implementing the various algorithms discussed in Chapter 6.*

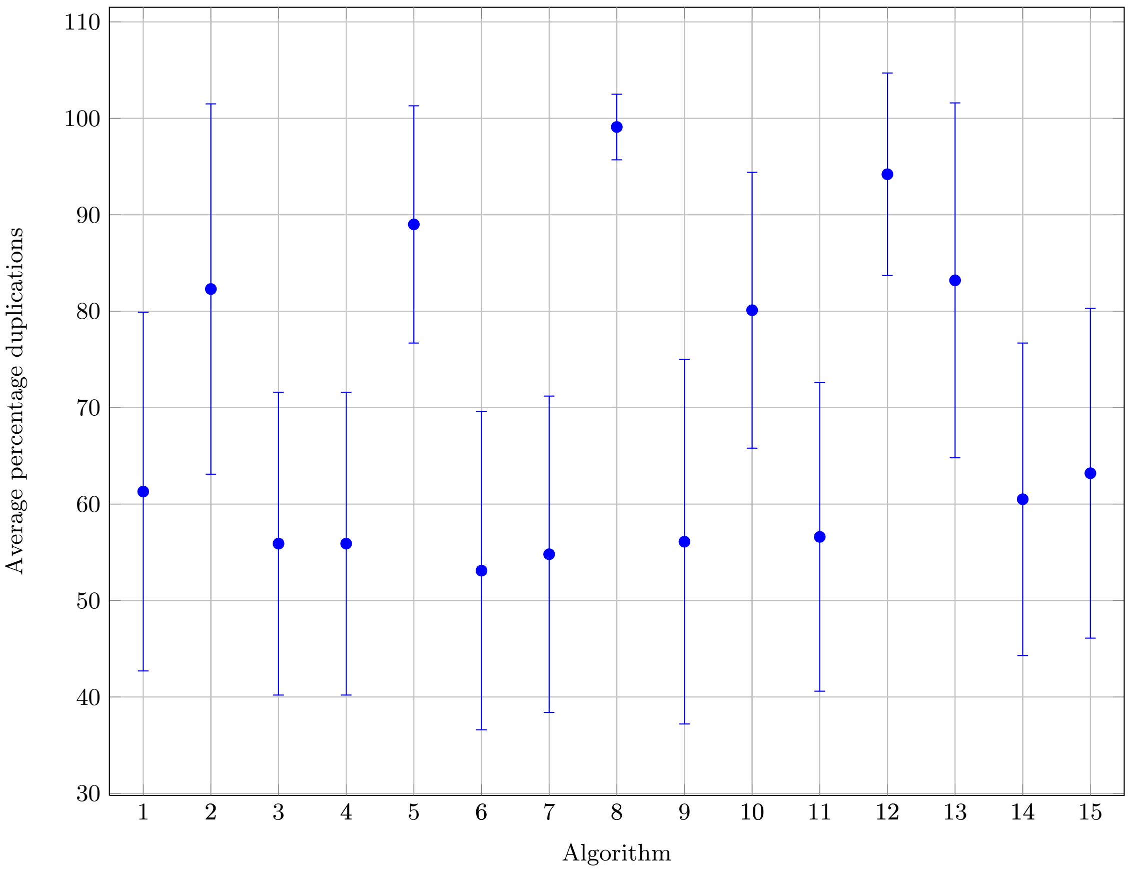


Figure 7.5: *The average percentage SKUs duplicated that yields the minimum number of cycles when implementing the various algorithms presented in Chapter 6 as in Table 7.2 and the standard deviation in the percentage duplications.*

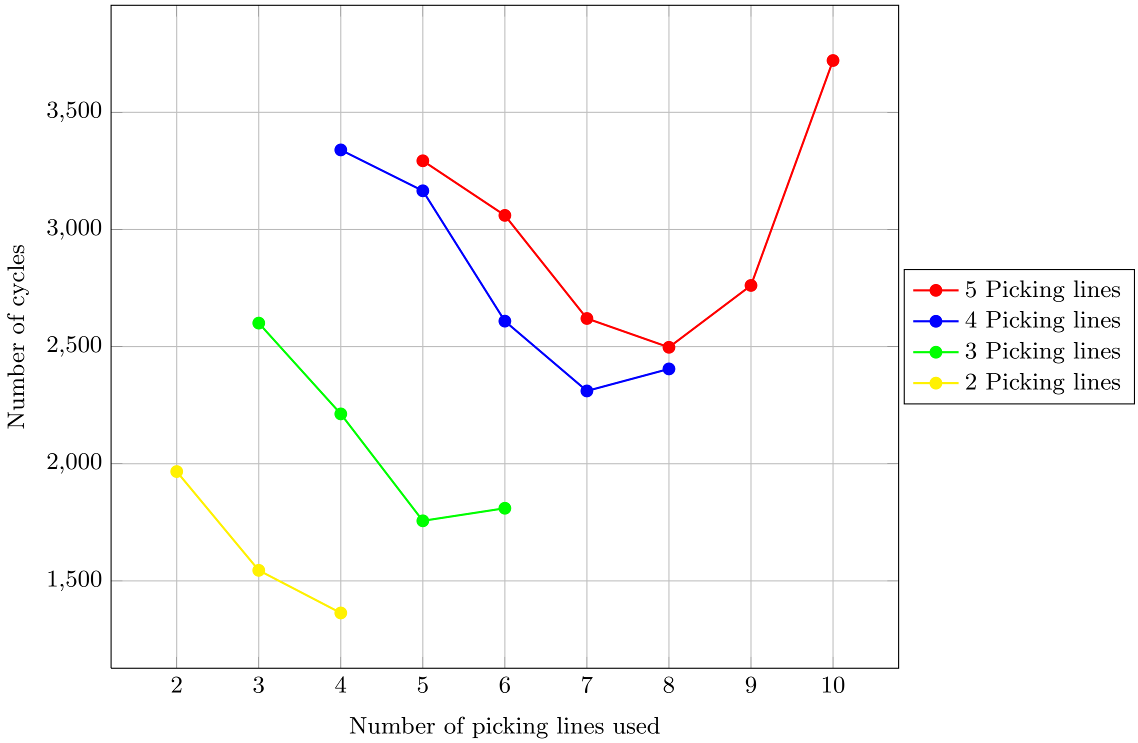


Figure 7.6: *Comparison of the average number of cycles to complete the branch orders for Algorithm 5 (PC/D/M2-S) with an increase in the number of picking lines as in Figure C.5, representing Group A1.*

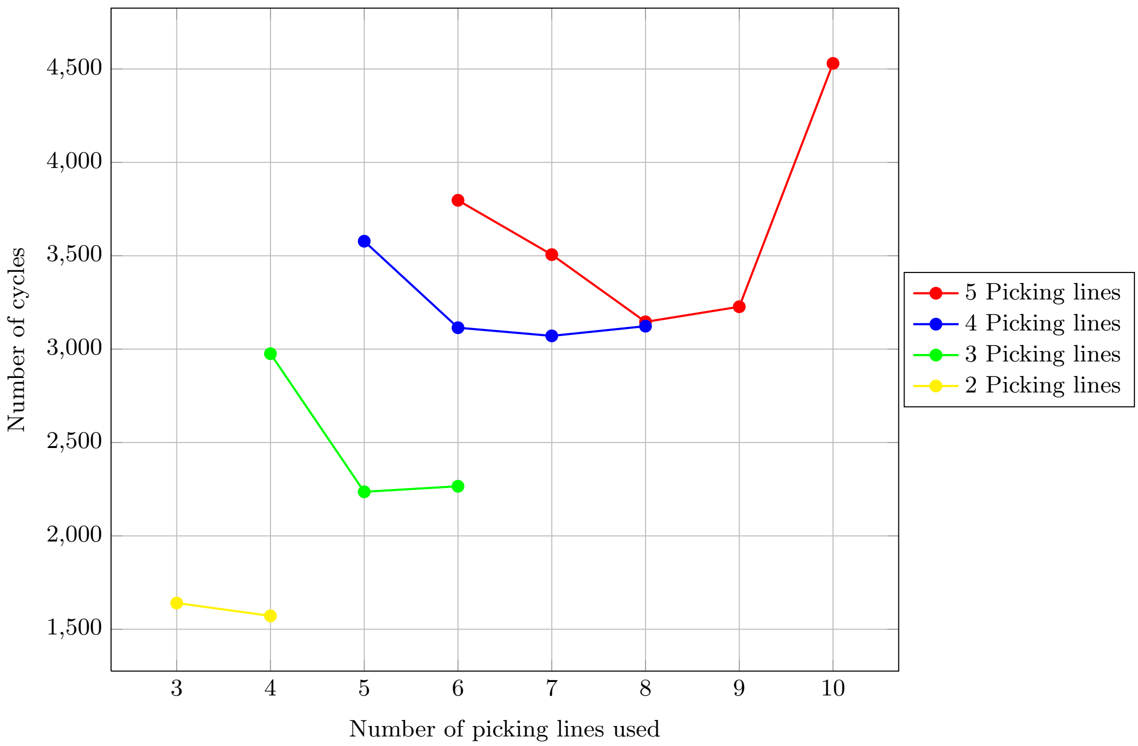


Figure 7.7: *Comparison of the average number of cycles to complete the branch orders for Algorithm 2 (PS/D/M1-S) with an increase in the number of picking lines as in Figure C.2, representing Group A2.*

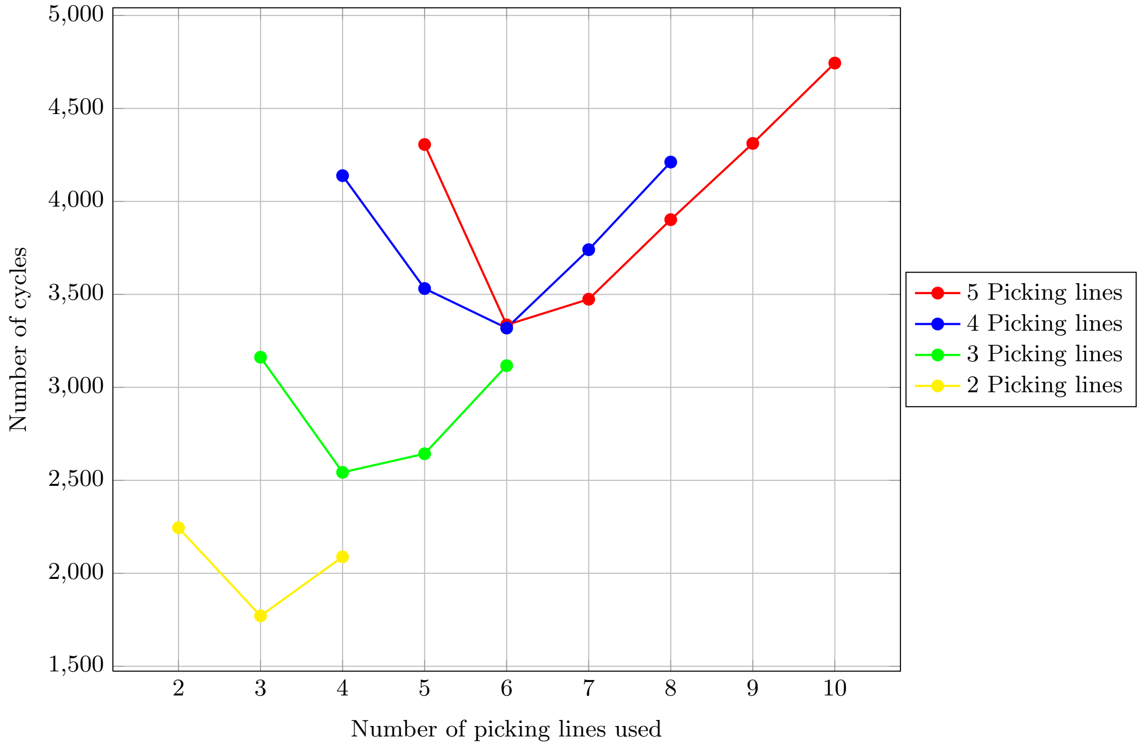


Figure 7.8: *Comparison of the average number of cycles to complete the branch orders for Algorithm 7 (PC/M1/D) with an increase in the number of picking lines as in Figure C.7, representing Group A3.*

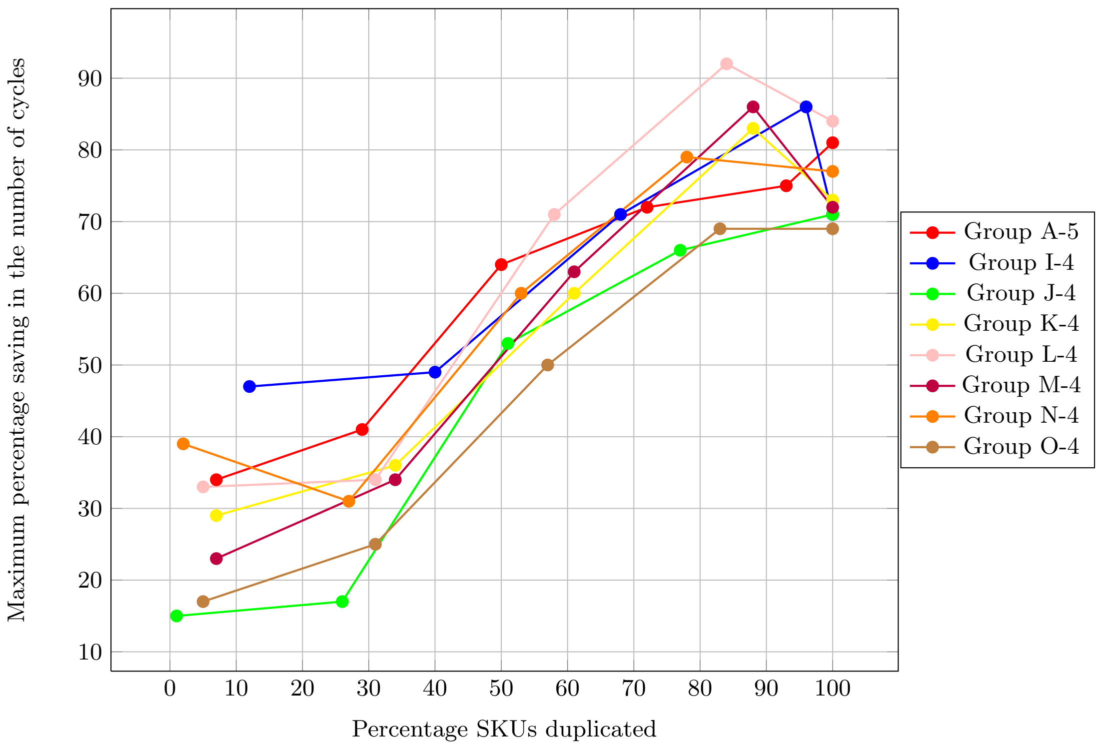


Figure 7.9: *A graph representing the maximum percentage savings achieved for the groups consisting of 4 or 5 of the original picking lines and the percentage SKUs duplicated yielding the maximum savings.*

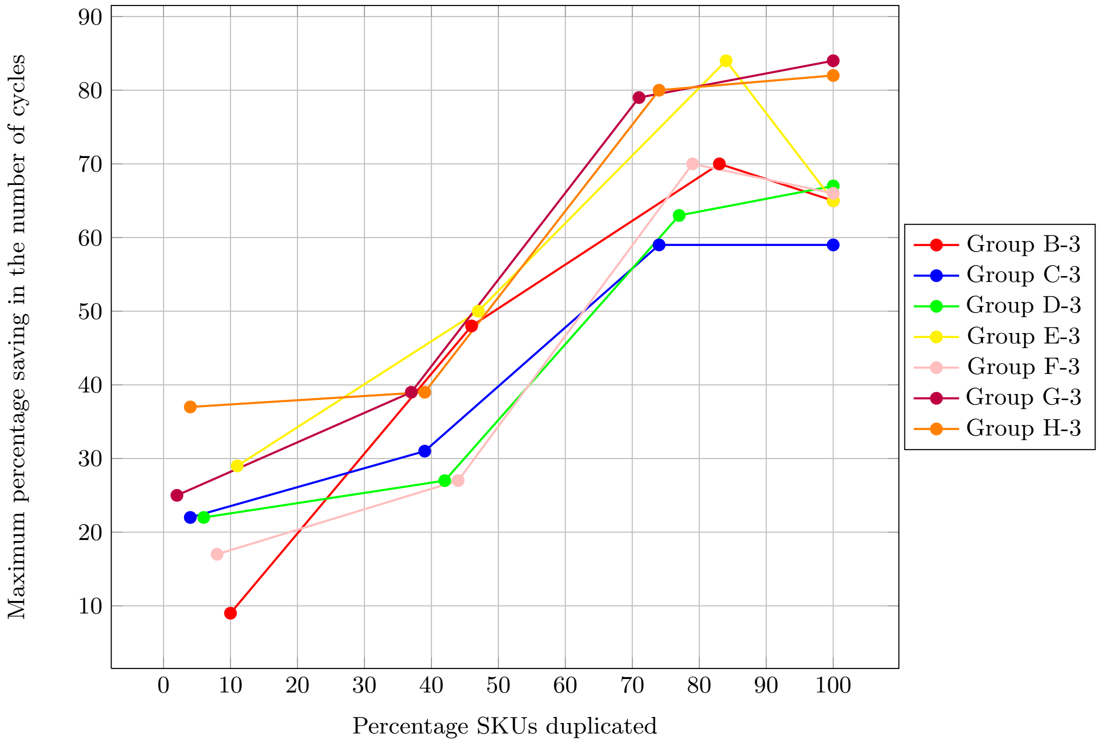


Figure 7.10: *A graph representing the maximum percentage savings achieved for the groups consisting of 3 of the original picking lines and the percentage SKUs duplicated yielding the maximum savings.*

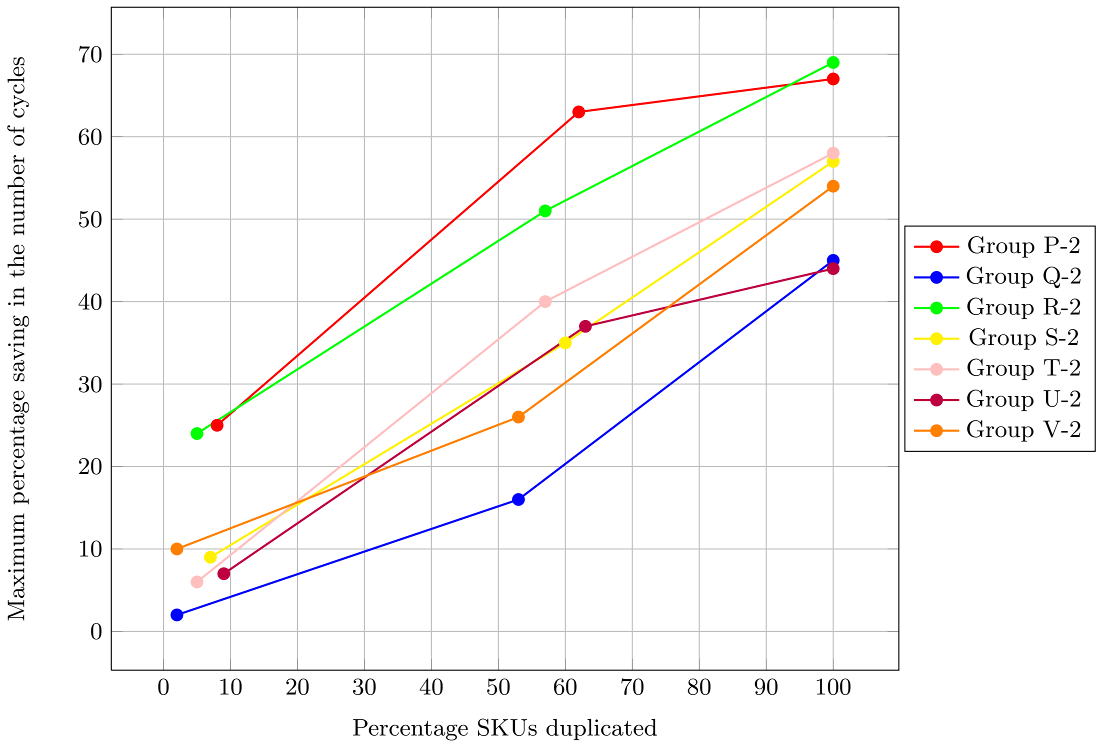


Figure 7.11: *A graph representing the maximum percentage savings achieved for the groups consisting of 2 of the original picking lines and the percentage SKUs duplicated yielding the maximum savings.*

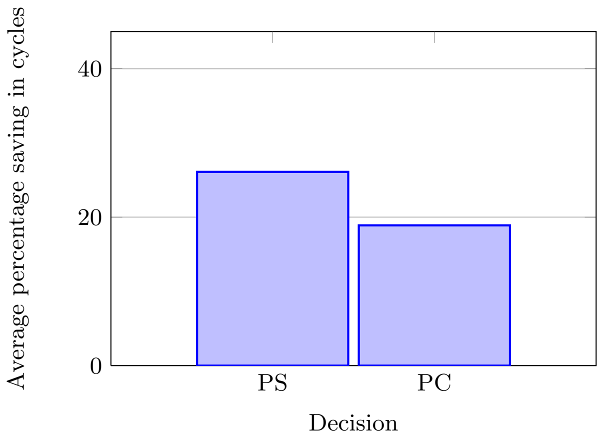


Figure 7.12: *Average percentage saving in the number of cycles for Algorithms 1 and 2 (PS) and Algorithms 3 and 4 (PC).*

gorithms implementing PS and PC. When comparing Algorithm 1 with Algorithm 3 and Algorithm 2 with Algorithm 4, for both cases the variance in the minimum number of cycles is greater than 300 cycles for more than 50% of the groups investigated in this study, as illustrated by Figure 7.17. There is thus a significant difference in the minimum number of cycles for the majority of the groups for these algorithms. The algorithms where the original picking lines are treated separately (PS, Algorithms 1 and 2) outperforms the algorithms where the original picking lines are combined (PC, Algorithms 3 and 4) for the majority of the cases with a significant variance in the minimum total number of cycles required to complete the order picking for the various groups.

When comparing the total number of cycles to determine the influence of duplicating or assigning SKUs first, Algorithms 4 is compared with Algorithm 7, Algorithm 5 and 8 as well as Algorithm 6 with Algorithm 9. Figure 7.13 give the average percentage saving for the algorithms implementing DA (Algorithms 4, 5 and 6) and for the algorithms implementing AD (Algorithms 7, 8 and 9). In these instances 16, 14 and 11 of the 22 groups has a variance in the minimum number of cycles of less than or equal to 50 cycles respectively, as illustrated in Figure 7.17. There is thus not a significant variance between the total number of cycles for the algorithms where the SKUs are first duplicated and then assigned to the picking lines and when they are first assigned to the picking lines and then duplicated and the 2 groups of algorithms outperforms each other almost an equal number of times. There is no real variance in the average percentage saving in the number of cycles. It can thus be concluded that duplicating or assigning SKUs first has no significant influence on the number of cycles required to complete the order picking for the groups investigated in this study.

To determine the influence of the various assignment methods on the minimum total number of cycles required to complete the order picking for all the groups, the various assignment methods is compared with each other by comparing the results for Algorithm 4 and 5 (M1 and M2), 7 and 8 (M1 and M2), 4 and 6 (M1 and M3), 7 and 9 (M1 and M3), 5 and 6 (M2 and M3) and 8 and 9 (M2 and M3). Figure 7.14 illustrate the average percentage saving in the number of cycles for the algorithms implementing M1 (Algorithms 4 and 7), M2 (Algorithms 5 and 8) and M3 (Algorithms 6 and 9). The average percentage saving for M2 is much higher than for M1 and M3. Thus there is no significant difference in the minimum total number of cycles when M1 and M3 is implemented, but M2 outperforms M1 and M3 for all the investigated groups in

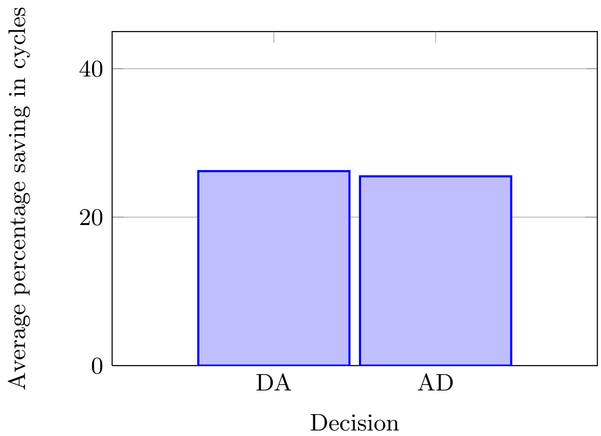


Figure 7.13: *Average percentage saving in the number of cycles for Algorithms 4, 5 and 6 (DA) and Algorithms 7, 8 and 9 (AD).*

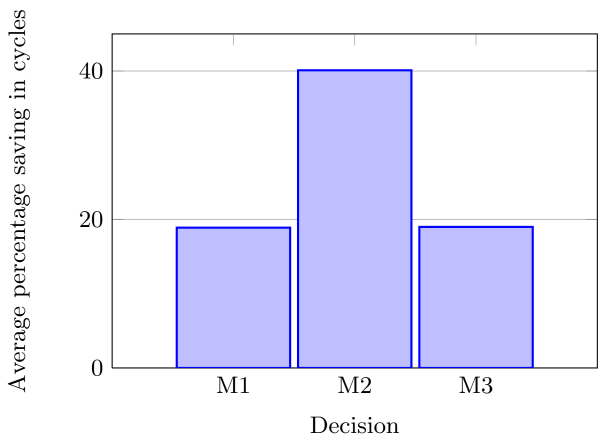


Figure 7.14: *Average percentage saving in the number of cycles for Algorithms 4 and 7 (M1), Algorithms 5 and 8 (M2) and Algorithms 6 and 9 (M3).*

terms of the minimum total number of cycles.

The influence of the various clustering methods are determined by comparing Algorithms 10, 11, 12 and 13, which all make use of the SLSS assignment (M2) as well as Algorithms 12, 13, 14 and 15 to determine the influence of the similarity and dissimilarity matrix clustering on the minimum total number of cycles. Figure 7.15 indicate the average percentage saving in the number of cycles, indicating that C1 and C3 yields the biggest average saving. Therefore clustering by means of the similarity matrix outperforms branch clustering and clustering by means of the similarity matrix (Algorithms 12 and 14) outperforms clustering by means of a dissimilarity matrix (Algorithms 13 and 15) approximately 2:1.

The final question in the decision flow discussed in §6.1, is whether the non-duplicate and duplicate SKUs are to be treated separately (ND) or simultaneously (S) when SKUs are assigned to the picking lines after being duplicated. For this Algorithms 1 (PS/D/M1-ND) and 2 (PS/D/M1-S) as well as Algorithms 3 (PC/D/M1-ND) and 4 (PC/D/M1-S) is compared. Figure 7.16 indicate the average percentage saving in the number of cycles when implementing ND and S. The variance in the minimum total number of cycles required to complete the order picking for the various groups between Algorithm 1 and 2 is more or less evenly distributed, as

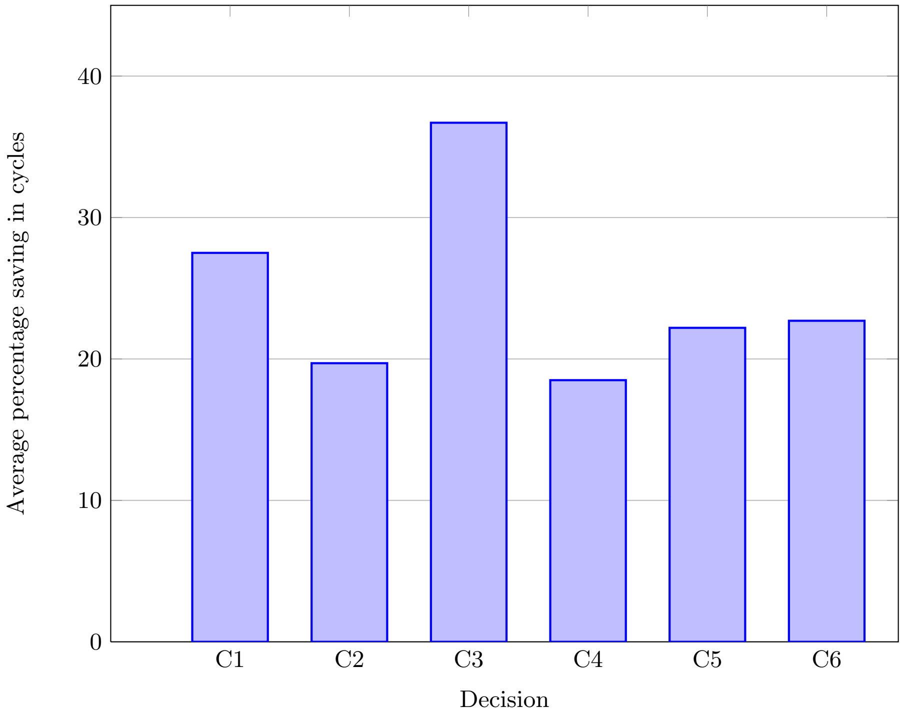


Figure 7.15: *Average percentage saving in the number of cycles for the clustering algorithms, Algorithms 10–15.*

illustrated by Figure 7.17. There is thus no significant difference in the number of cycles when implementing ND or S.

7.5 Work balance

From the previous sections it is clear that some of the decision flow factors implemented by the algorithms influence the minimum number of cycles to complete the order picking more than others. The assignment method, mainly M2, or clustering method implemented has the biggest

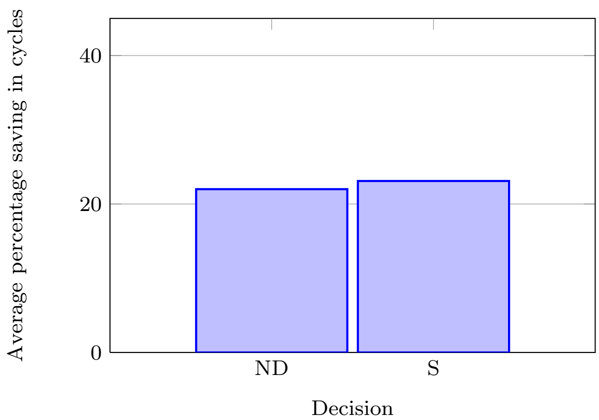


Figure 7.16: *Average percentage saving in the number of cycles for Algorithms 1 and 3 (ND) and Algorithms 2 and 4 (S).*

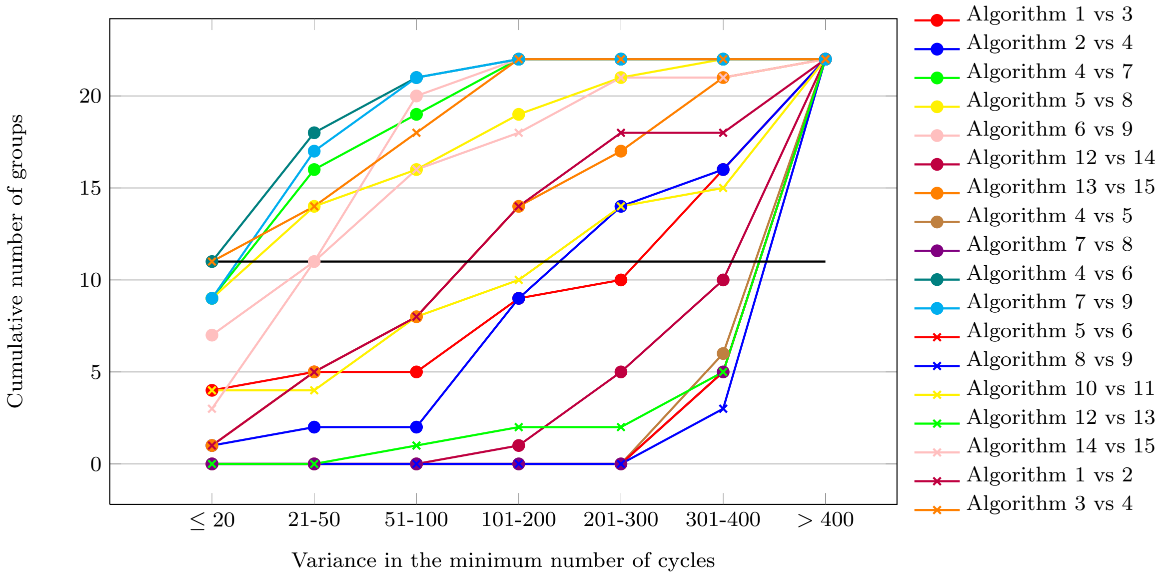


Figure 7.17: *A graph representing the cumulative number of groups and the corresponding absolute variance in the number of cycles for the compared algorithms as indicated based on the decision flow of the various algorithms as discussed in Chapter6.*

influence on the number of cycles that the picking lines require to complete the order picking. This is also the factors that influence the work balance over the various picking lines. A decrease in the work balance over the picking lines, increase the variance in the number of cycles for each of the picking lines. Figure 7.18 represent the average percentage saving in the number of cycles for the algorithms that have a poor, average and good work balance, Groups A1, A2 and A3 respectively.

There is no significant difference between the number of cycles required by the picking lines for assignment method 1 and 3, or when SKUs are first duplicated or first assigned, or when non-duplicate and duplicate SKUs are treated seperately or simultaneously. This is mostly due to the fact that these decisions does not cause an unbalanced work distribution over the picking lines. From Figure 7.12 it is clear that PS slightly outperforms PC, this is mostly due to the fact that the work balance in PC is better than for PS. M1 and M3 also has a more or less equal work balance which is much better than for M2. From this section it may be concluded that if the decisions implemented influence the work balance distribution over the various picking lines negatively, a greater saving is achieved in the number of cycles that is required to complete the order picking.

7.6 Chapter conclusion

PS slightly outperforms PC in terms of the number of cycles that is required to complete the order picking but also require that approximately 15% more SKUs are duplicated to achieve the minimum number of cycles. This saving in the number of cycles is mostly because of the unbalanced work distribution over the individual picking lines as the algorithms implementing PC enforce a slightly better work balance than the algorithms implementing PS. There are no significant difference between the percentage cycles saved or the percentage of SKUs that are duplicated to achieve the minimum number of cycles when the SKUs are first duplicated and

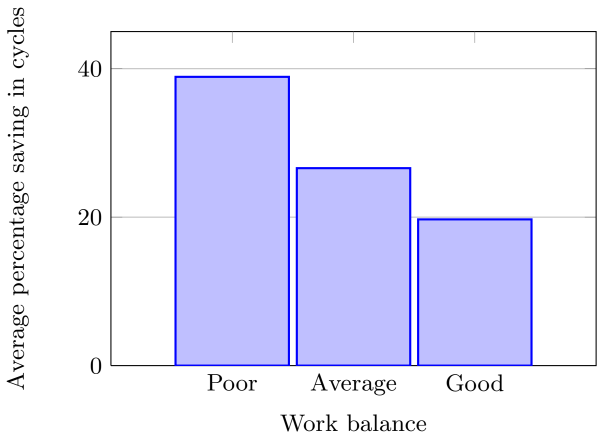


Figure 7.18: *The average percentage saving in the number of cycles for the algorithms implementing a poor work balance amongst the various picking lines (Group A1, Algorithms 5, 8 and 12), as well as for algorithms implementing an average work balance (Group A2, Algorithms 1, 2 and 10) and a good work balance (Group A3, Algorithms 3, 4, 6, 7, 9, 11, 13, 14 and 15).*

then assigned to the various picking lines or when they are first assigned to the picking lines and then duplicated.

For the assignment methods, there are no significant difference between M1 and M3 in terms of the minimum number of cycles or the number of SKUs duplicated to achieve the minimum cycles. This is the case because both assignment methods implement a form of cyclical assignment. M2 on the other hand achieve a significantly bigger saving in the number of cycles than M1 and M3, but also require that approximately 30% more of the SKUs are duplicated than M1 and M3 to achieve the minimum number of cycles.

For the branch clustering algorithms there are a small variance in the minimum number of cycles with Algorithm 10 requiring that approximately 25% more SKUs are duplicated than for Algorithm 11 to achieve the minimum number of cycles. There are no significant difference in the minimum number of cycles for the SKU clustering algorithms (Algorithms 12–15), but the opposite is true for the percentage SKUs that are duplicated at the minimum number of cycles with Algorithm 12 outperforming the others. This is mostly due to the fact that Algorithm 12 and 13 implements M2 while Algorithms 14 and 15 implements M1. Furthermore, there is no significant difference between the minimum number of cycles or the number of duplications at which the minimum cycles is achieved for algorithms implementing ND or S strategy when duplicating.

As Algorithms 5 and 8 outperforms the other algorithms in terms of the minimum number of cycles, but also require the greatest percentage SKUs to be duplicated to achieve the minimum number of cycles, it may be concluded that the poorer the work balance is distributed over the individual picking lines the bigger the saving in the number of cycles it would take to complete the order picking. Also a greater percentage of SKUs needs to be duplicated in order to achieve this saving. Furthermore duplicating first or assigning the SKUs first has no significant influence on the number of cycles which is also true for the case when the non-duplicate and duplicate SKUs are treated separately or simultaneously when assigning them to the picking lines.

CHAPTER 8

Conclusions

Contents

This chapter gives a brief summary of the work done in this thesis. The final conclusions and remarks are also given as well as some ideas for future research.

8.1 Summary of findings

As discussed in Chapter 4, the questions that need to be answered in this study is:

1. Which SKUs to duplicate?

   From the results discussed in Chapter 5 it is clear that the SKUs with the highest pick frequency should be duplicated first as the pick frequency of the SKUs forms a lower bound on the number of cycles it would take the pickers to complete the order picking. It was also illustrated that by assigning duplicate SKUs to the unused locations even more savings are possible in terms of the travel distance of pickers. Furthermore, with an increase in the number of locations required by the SKUs and therefore also an increase in the distance of each cycle, there is a decrease in the number of cycles that the pickers have to complete to make all the picks. Therefore there is a trade-off between the number of SKUs to duplicate and the travel distance per cycle for the pickers. The conclusion is also made that when 2 SKUs are duplicated (the 2 SKUs with the highest pick frequency on the picking line), 2 SKUs are required to be removed from the picking line in order to maintain an equal number of locations required by the SKUs on the picking line, then removing the 2 SKUs with the next highest pick frequency bring about the greatest saving in the number of cycles the order pickers will have to complete. This again formed the basis of Algorithm 1 in Chapter 6. It was also concluded that it would be too time consuming to try every possible combination of duplicating and removing SKUs in order to determine the combination of duplications and removing SKUs that would yield the minimum number of cycles. Thus these findings formed the basis for the algorithms in

Chapter 6 that was suggested in order to decrease the number of cycles by duplicating and assigning SKUs to picking lines.

2. Number of SKUs to duplicate?

   For some of the groups only a small saving in the number of cycles was achieved when only a few SKUs were duplicated. On the other hand, there was a decrease in the savings in cycles for some of the groups when all the SKUs were duplicated. It is thus concluded that by duplicating 40–70% of the SKUs on a picking line a very significant saving in the total number of cycles is achieved.

3. Which SKUs to remove from the line?

   This question becomes irrelavant because a better work balance is achieved by implementing PC, which is how PEP is currently assigning SKUs to picking lines.

4. How to assemble the new lines?

   Three assignment methods was identified, cyclical assignment (M1), set length subset sequential assignment (SLSS, M2) and remaining high, low cyclical assignment (RHLC, M3). From an optimisation point of view M2 always outperformed M1 and M3 by a significant number of cycles but also require that more SKUs are duplicated to achieve the minimum number of cycles and produces a poor work balance. There is no significant difference between M1 and M3 and both has a very good work balance.

5. Duplicate SKUs before or after assigning them to picking lines?

   The conclusion was made that there is no difference between the two approaches and that they yield equal savings in terms of the number of cycles for all the groups investigated in this study.

The conclusion is made that the more unbalanced the workload is distributed over the individual picking lines and the greater the number of duplicated SKUs, the greater the possible saving in the total number of cycles that is required to complete all the order picking.

8.2 Recommendations to PEP

PEP strive to have a balanced workload over the various picking lines. M2 is thus not preferred by PEPs management, because of its poor work balance. From the results discussed in Chapter 7 it is recommended to PEP to continue working from a combined DBN list when they start assigning SKUs to picking lines. Furthermore 32 to 40 SKUs has to be assigned to a picking line by means of cyclical (M1) or RHLC assignment (M3) as these assignment methods create a more balanced work load over the picking lines. The picking lines are then filled by duplicating the SKUs with the highest pick frequencies on each of the picking lines. This will yield an average saving in the number of cycles that is required to complete the order picking by approximately 20%. Therefore implementing either Algorithm 7 (PC/M1/D) or 9 (PC/M3/D) is recommended as they also have a shorter runtime than the clustering algorithms.

8.3 Future research

Some ideas for future work include:

1. Investigating the influence that spreading SKU's and their duplicates over more than 1 picking line has on the number of cycles that would be required to complete the order picking to determine if a further saving is possible.

2. To determine if further savings in the number of cycles is possible by assigning the duplicated SKUs to more than 2 locations on a single or multiple picking lines.

3. Analyse the picking line profile to determine the exact number of SKUs to duplicate that would yield the biggest saving in the number of cycles that is required to complete the order picking and if there are similarities in the profile at which the biggest saving in the number of cycles is achieved.

4. Determine what factors influence the workload on a picking line and how to assign the various SKUs to picking lines in order to balance the workload equally as this is essential for PEPs management.

5. Determine what the effect of the location of the SKUs on a picking line is when SKUs are duplicated, on the number of cycles that is required to complete the order picking.

8.4 Objectives achieved

In §1.2 the following objectives were identified:

Objective I: To describe the layout and operations of the DC so that the problem may be viewed in the broader DC context;

Objective II: To describe the order picking system in detail so that the characteristics of the problem may be understood;

Objective III: Identify constraints and make suitable assumptions so that a detailed problem may be identified and modelled;

Objective IV: Present multiple duplicating algorithms that construct picking lines to compare the results, in terms of the number of cycles it would take the pickers to complete the order picking;

Objective V: Discuss potential directions of future studies.

In Chapters 1 and 3 Objectives I and II was achieved. The operations within a DC in general was discussed in Chapter 1 as well as an overview of the supply chain. Chapter 3 discussed the layout and operations of PEP's Durban DC with a thorough explanation of the order picking system. In §3.6 certain restrictions was made mainly because of the systems implemented by PEP and managerial decisions imposed by the warehouse management team. These constraints and the experimetal results discussed in Chapter 5 lead to the assumtions made in §5.4. This was done in fulfilment of Objective III. Objective IV was achieved in Chapter 6, where 15 algorithms was discussed. The results of these algorithms are discussed and compared in Chapter 7. Finally Objective V is achieved in §8.3 where future work is suggested.

References

[1] Abdel-Malek L & Tang C, 1994, *A heuristic for cyclic stochastic sequencing of tasks on a drum-like storage system*, Computers Operations Research, **21 (4)**, pp. 385–396.

[2] Accounting for management, *Kanbans*, [Online], [Cited June 26th, 2012], Available from `http://www.accounting4management.com/kanbans.htm`

[3] Anderson DL, Britt FF & Favre DJ, 2007, *The seven pinciples of supply chain management*, Supply Chain Management Review, pp. 3–8.

[4] Armbruster D, Gel ES & Murakami J, 2007, *Bucket brigades with worker learning*, European Journal Of Operational Research, **176**, pp. 264–274.

[5] Ashayeri J & Gelders LF, 1985, *Warehouse design optimization*, European Journal Of Operational Research, **21**, pp. 285–294.

[6] Barloworld Logistics, 2004, *Redesign, implementation and ongoing management of outbound distribution for PEP stores*, [Online], [Cited June 24th, 2011], Available from `http://www.barloworld-logistics.com/wp-content/uploads/2012/04/PEP-Case-Study.pdf`

[7] Bartholdi JJ, Bunimovich LA & Eisenstein DD, 1999, *Dynamics of two- and three-worker "bucket brigade" production lines*, Operations research, **47 (3)**, pp. 488–492.

[8] Bartholdi JJ & Eisenstein DD, 1996, *A production line that balances itself*, Operations Research, **44 (1)**, pp. 21–34.

[9] Bartholdi JJ & Eisenstein DD, 2005, *Using bucket brigades to migrate from craft manufacturing to assembly lines*, Manufacturing and Service Operations Management, **7 (2)**, pp. 121–129.

[10] Bartholdi JJ, Eisenstein DD & Foley RD, 2001, *Performance of bucket brigades when work is stochastic*, Operations Research, **49 (5)**, pp. 710–719.

[11] Bartholdi JJ, Eisenstein DD & Lim YF, 2006, *Bucket brigades on in-tree assembly networks*, European Journal Of Operational Research, **168**, pp. 870–879.

[12] Bartholdi JJ, Eisenstein DD & Lim YF, 2009, *Deterministic chaos in a model of discrete manufacturing*, Naval Research Logistics, **56**, pp. 293–299.

[13] Bartholdi JJ & Gue KR, 2000, *Reducing labor costs in an LTL crossdocking terminal*, Operations Research, **48 (6)**, pp. 823–832.

[14] Bartholdi JJ & Hackman ST, 2008, *Allocating space in a forward pick area of a distribution center for small parts*, IIE Transactions, **40**, pp. 1046–1053.

[15] Bartholdi JJ & Hackman ST, 2008, *Warehouse & distribution science*, [Online], [Cited June 24th, 2011] Available from `http://www.warehouse-science.com/`

[16] Bartholdi JJ & Hackman ST, 2010, *Warehouse & distribution science*, [Online], [Cited June 24th, 2011] Available from `http://www.warehouse-science.com/`

[17] Braam ITJ, Van Dormolen M & Frings-Dresen MHW, 1996, *The work load of warehouse workers in three different working systems*, International Journal of Industrial Ergonomics, **17**, pp. 469–480.

[18] Bullas J, 2011, *50 Amazing facts and figures about Google*, [Online], [Cited June 24th, 2011], Available from `http://www.jeffbullas.com/2011/05/16/50-amazing-facts-and-figures-about-google/`

[19] Caputo AC & Pelagagge PM, 2006, *Management criteria of automated order picking systems in high-rotation high-volume distribution centers*, Industrial Management and Data Systems, **106 (9)**, pp. 1359–1383.

[20] Chan FTS & Chan HK, 2011, *Improving the productivity of order picking of a manual-pick and multi-level rack distribution warehouse through the implementation of class-based storage*, Expert Systems With Applications, **38**, pp. 2686–2700.

[21] Chew EP & Tang LC, 1999, *Travel time analysis for general item location assignment in a rectangular warehouse*, European Journal Of Operational Research, **112**, pp. 582–597.

[22] Clarke G & Wright JW, 1964, *Scheduling of vehicles from a central depot to a number of delivery points*, Operations research, **12 (4)**, pp. 568–581.

[23] Conveyco, *Textile industry*, [Online], [Cited June 24th, 2011], Available from `http://www.conveyco.co.za/textile-industry.htm`

[24] Daniels RL, Rummel JL & Schantz R, 1998, *A model for warehouse order picking*, European Journal Of Operational Research, **105**, pp. 1–17.

[25] Dekker R, De Koster MBM, Roodbergen KJ & Van Kalleveen H, 2004, *Improving order-picking response time at Ankor's warehouse*, Interfaces, **34 (4)**, pp. 303–313.

[26] De Koster MBM, Le-Duc T & Roodbergen KJ, 2007, *Design and control of warehouse order picking : A literature review*, European Journal Of Operational Research, **182**, pp. 481–501.

[27] De Koster MBM, Van Der Poort ES & Wolters M, 1999, *Efficient orderbatching methods in warehouses*, International Journal of Production Research, **37 (7)**, pp. 1479–1504.

[28] De Villiers AP, 2011, *Minimising the total travel distance to pick orders on a unidirectional picking line*, MSc Thesis, University of Stellenbosch, Stellenbosch.

[29] De Villiers AP & Visagie SE, 2012, *Toewysingsheuristieke om die volgorde van bestellings vir 'n uitsoeklyn te bepaal*, LitNet Akademies, **9 (1)**, pp. 1–22.

[30] De Villiers AP, Matthews J & Visagie SE, 2012, *Tour construction heuristics for an order sequencing problem*, South African Journal of Industrial Engineering, **23 (3)**, pp. 56–67.

[31] Dorigo M & Gambardella LM, 1997, *Ant colonies for the travelling salesman problem*, Biosystems, **43**, pp. 73–81.

[32] Dorigo M & Gambardella LM, 1997, *Ant colony system: A cooperative learning approach to the traveling salesman problem*, IEEE Transactions on Evolutionary Computation, **1 (1)**, pp. 53–66.

[33] Dukic G & Oluic C, 2005, *Order-picking methods: Improving order-picking efficiency*, Paper presented at the International Conference on Industrial Logistics 2005, Montevideo

[34] EDS Engineering, 2007, *PEPKOR Warehouse: Durban*, [Online], [Cited June 24$^{\text{th}}$, 2011], Available from `http://www.edseng.co.za/warehouse-pepkor-warehouse.php`

[35] Egbelu PJ, 1993, *A comparison of dwell point rules in an AS/RS*, International Journal of Production Research, **31 (11)**, pp. 2515–2530.

[36] Eisenstein DD, 2008, *Analysis and optimal design of discrete order picking technologies along a line*, Naval Research Logistics, **55 (4)**, pp. 350–362.

[37] Ellis Systems Corporation, , *Horizontal carousels*, [Online], [Cited June 12$^{\text{th}}$, 2012], Available from `http://www.ellismh.com/index.php?url=horizontal-carousels`

[38] Ellis Systems Corporation, , *Vertical carousels*, [Online], [Cited June 12$^{\text{th}}$, 2012], Available from `http://www.ellismh.com/index.php?url=vertical-carousels`

[39] Elsayed EA, 1981, *Algorithms for optimal material handling in automatic warehousing systems*, International Journal of Production Research, **19 (5)**, pp. 525–535.

[40] Elsayed EA & Stern, 1983, *Computerized algorithm for order processing in automated warehousing systems*, International Journal of Production Research, **21**, pp. 579–586.

[41] Elsayed EA & Unal OI, 1989, *Order batching algorithms and travel-time estimation for automated storage/retrieval systems*, International Journal of Production Research, **27 (7)**, pp. 1097–1114.

[42] Everitt BS, Landau S, Leese M & Stahl D, 2011, *Cluster analysis*, 5$^{\text{th}}$ Edition, John Wiley & Sons, Kings College London (UK).

[43] Faber N, De Koster MBM & Van De Velde SL, 2007, *Linking warehouse complexity to warehouse planning and control structure*, International Journal of Physical Distribution and Logistics Management, **32 (5)**, pp. 381–395.

[44] Fa-liang C, Zeng-Xiao L, Zheng X & Dong-Dong L, 2007, *Research on order picking optimization problem of automated warehouse*, Systems Engineering - Theory and Practice, **27 (2)**, pp. 139–143.

[45] Gademann N & Van De Velde S, 2005, *Order batching to minimize total travel time in a parallel-aisle warehouse*, IIE Transactions, **37**, pp. 63–75.

[46] Gagliardi J-P, Ruiz A & Renaud J, 2008, *Space allocation and stock replenishment synchronization in a distribution center*, International Journal Production Economics, **115**, pp. 19–27.

[47] Ganeshan R & Harrison TP, 1995, *An introduction to supply chain management*, [Online], [Cited June 24th, 2011], Available from http://lcm.csa.iisc.ernet.in/scm/supply_chain_intro.html

[48] Gelders LF, 1992, *Warehouse design and operation*, European Journal Of Operational Research, **58**, pp. 1–2.

[49] Ghosh JB & Wells CE, 1992, *Optimal retrieval strategies for carousel conveyors*, Mathematical Computer Modelling, **16 (10)**, pp. 59–70.

[50] Goetschalckx M & Ratliff HD, 1988, *An efficient algorithm to cluster order picking items in a wide aisle*, Engineering Costs and Production Economics, **13**, pp. 263–271.

[51] Gong Y, 2009, *Stochastic modelling and analysis of warehouse operations*, PhD Thesis, Erasmus University Rotterdam, Nederland.

[52] Gong Y & De Koster MBM, 2008, *A polling-based dynamic order picking system for online retailers*, IE Transactions, **40**, pp. 1070–1082.

[53] Gu J, Goetschalckx M & McGinnis LF, 2010, *Research on warehouse design and performance evaluation: A comprehensive review*, European Journal of Operational Research, **203**, pp. 539–549.

[54] Gu J, Goetschalckx M & McGinnis LF, 2007, *Research on warehouse operation: A comprehensive review*, European Journal of Operational Research, **177**, pp. 1–21.

[55] Ha J-W & Hwang H, 1994, *Class-based storage assignment policy in carousel system*, Computers & Industrial Engineering, **26 (3)**, pp. 489–499.

[56] Hall RW, 1993, *Distance approximation for routing manual pickers in a warehouse*, IIE Transactions, **25**, pp. 77–78.

[57] Hassini E, 2009, *One-dimensional carousel storage problems: Applications, review and generalizations*, Infor, **47 (2)**, pp. 81–92.

[58] Helsgaun K, 2000, *An effective implementation of the LinKernighan traveling salesman heuristic*, European Journal of Operations Research, **126**, pp. 106–130.

[59] Ho SS & Sarma S, 2008, *The fragmented warehouse: Location assignment for unit-load picking*, Proceedings of the IEEE International Conference on Industrial Engineering and Engineering Management 2008, Singapore, pp. 1159–1163.

[60] Hollingsworth BK, 2003, *Decision strategy to minimize replenishment costs in a distribution center with forward-reserve storage*, MSc Thesis, Frits J. and Dolores H. Russ College of Engineering and Technology, Ohio University.

[61] Hsieh L-F & Huang Y-C, 2011, *New batch construction heuristics to optimise the performance of order picking systems*, International Journal of Production Economics, **131**, pp. 618–630.

[62] Hsu C-M, Chen K-Y & Chen M-C, 2005, *Batching orders in warehouses by minimizing travel distance with genetic algorithms*, Computers in Industry, **56**, pp. 169–178.

[63] Hwang H, Baek W & Lee M-K, 1988, *Clustering algorithms for order picking in an automated storage and retrieval system*, International Journal of Production Research, **26 (2)**, pp. 189–201.

[64] Hwang HS & Cho GS, 2006, *A performance evaluation model for order picking warehouse design*, Computers & Industrial Engineering, **51**, pp. 335–342.

[65] Hwang H & Lee M-K, 1988, *Order batching algorithms for a man-on-board automated storage and retrieval system*, Engineering Costs and Production Economics, **13**, pp. 285–294.

[66] Jane C-C & Laih Y-W, 2005, *A clustering algorithm for item assignment in a synchronized zone order picking system*, European Journal Of Operational Research, **166**, pp. 489–496.

[67] JDA Software, 2010, *South African retailer streamlines merchandise planning processes, decreases overstocks and improves profitability*, [Online], [Cited June 24th, 2011], Available from `http://www.jda.com/realresultsmagazine/view-article.cfm?did=1635`

[68] Jewkes E, Lee C & Vickson R, 2004, *Product location, allocation and server home base location for an order picking line with multiple servers*, Computers & Operations Research, **31**, pp. 623–636.

[69] Kallina C & Lynn J, 1976, *Application of the cube-per-order index rule for stock location in a distribution warehouse*, Interfaces, **7 (1)**, pp. 37–46.

[70] Kanet JJ & Ramirez RG, 1986, *Optimal stock picking decisions in automatic storage and retrieval systems*, Omega International Journal of Management Science, **14 (3)**, pp. 239–244.

[71] Kim KH, 1993, *A joint determination of storage locations and space requirements for correlated items in a miniload AS/RS*, International Journal of Production Research, **31 (11)**, pp. 2649–2659.

[72] Kong C & Masel D, 2008, *Methods for design and management of a fast-pick area in a warehouse*, Proceedings of the 2008 Industrial Engineering Research Conference, Vancouver, pp. 1367–1372.

[73] Lee CC, 2006, *Two-warehouse inventory model with deterioration under FIFO dispatching policy*, European Journal of Operational Research, **174**, pp. 861–873.

[74] Lin C-H & Lu I-Y, 1999, *The procedure of determining the order picking strategies in distribution center*, Operations research, **60-61**, pp. 301–307.

[75] Litvak N, 2006, *Optimal picking of large orders in carousel systems*, Operations Research Letters, **34**, pp. 219–227.

[76] Litvak N & Adan I, 2002, *On a class of order pick strategies in paternosters*, Operations Research Letters, **30**, pp. 377–386.

[77] Litvak N & Adan I, 2001, *The travel time in carousel systems under the nearest item heuristic*, Journal of Applied Probability, **38 (1)**, pp. 45–54.

[78] Litvak N, Adan I, Wessels J & Zijm WHM, 2000, *The nearest item heuristic for carousel systems*, [Online], [Cited June 24th, 2011], Available from `http://alexandria.tue.nl/repository/books/533218.pdf`

[79] Litvak N, Adan IJBF, Wessels J & Zijm WHM, 2001, *Order picking in carousel systems under the nearest item heuristic*, Probability in the Engineering and Informational Sciences, **15**, pp. 135–164.

[80] Litvak N & Vlasiou M, 2010, *A survey on performance analysis of warehouse carousel systems*, Statistica Neerlandica, **64 (4)**, pp. 401–447.

[81] Macro JG & Salmi RE, 2002, *A simulation tool to determine warehouse efficiencies and storage allocations*, Proceedings of the 2002 Winter Simulation Conference, San Diego, pp. 1274–1281.

[82] Malmborg CJ, 1996, *Storage assignment policy tradeoffs*, International Journal of Production Research, **34 (2)**, pp. 363–378.

[83] Malmborg CJ & Altassan KM, 1998, *Analysis of storage assignment policies in less than unit load warehousing systems*, International Journal of Production Research, **36 (12)**, pp. 3459–3475.

[84] Malmborg CJ, Balachandran S & Kyle DM, 1986, *A model based evaluation of a commonly used rule of thumb for warehouse layout*, Applied Mathematical Modelling, **10**, pp. 133–138.

[85] Malmborg CJ, Krishnakumar B & Simons GR, 1988, *A mathematical overview of warehousing systems with single/dual order-picking cycles*, Applied Mathematical Modelling, **12**, pp. 2–8.

[86] Manzini R, Gamberi M, Persona A & Regattieri A, 2007, *Design of a class based storage picker to product order picking system*, International Journal of Advanced Manufacturing Technology, **32**, pp. 811–821.

[87] Maserole WH, 1949, *Organization of wholesale operations for low cost*, The Journal of Marketing, **14 (2)**, pp. 192–197.

[88] Matthee DJP, 2011, *Bin size allocation on a picking line*, HonsBComm project, University of Stellenbosch, Stellenbosch.

[89] Matthews J, 2009, *Single fixed crane optimisation within a distribution centre*, HonsBComm project, University of Stellenbosch, Stellenbosch.

[90] Matthews J, 2011, *Order sequencing and SKU arrangement on a unidirectional picking line*, MSc Thesis, University of Stellenbosch, Stellenbosch.

[91] Matthews J & Visagie SE, 2011, *Single fixed crane optimisation within a distribution centre*, South African Journal of Industrial Engineering, **22 (2)**, pp. 107–120.

[92] Melachrinoudis E & Min H, 2007, *Redesigning a warehouse network*, European Journal of Operational Research, **176**, pp. 210–229.

[93] Meller RD C & Pazour JA, 2008, *A Heuristic for SKU assignment and allocation in an A-frame system*, Proceedings of the 2008 Industrial Engineering Research Conference, Vancouver, pp. 770–775.

[94] Molnar B & Lipovszki G, 2002, *Multi-objective routing and scheduling of order pickers in a warehouse*, International Journal of Simulation, **6 (5)**, pp. 22–32.

[95] Murphy PR & Wood DF, 2008, *Contemporary Logistics*, 9th Edition, Pearson Prentice Hall, Upper Saddle River (New Jersey)

[96] Netbeans IDE 6.8, [Online], [Cited November 1st, 2012], Available from `http://www.netbeans.org`.

[97] Owyong M & Yih Y, 2006, *Picklist generation algorithm with order-consolidation consideration for split-case module-based fulfilment centres*, International Journal of Production Research, **44 (21)**, pp. 4529–4550.

[98] Pan C-H & Liu S-Y, 1995, *A comparative study of order batching algorithms*, Omega International Journal of Management Science, **23 (6)**, pp. 691–700.

[99] Pan JC-H & Shih P-H, 2008, *Evaluation of the throughput of a multiple-picker order picking system with congestion consideration*, Computers & Industrial Engineering, **55**, pp. 379–389.

[100] Pan JC-H & Wu M-H, 2009, *A study of storage assignment problem for an order picking line in a pick-and-pass warehousing system*, Computers & Industrial Engineering, **57**, pp. 261–268.

[101] Parikh PJ & Meller RD, 2010, *A travel-time model for a person-onboard order picking system*, European Journal of Operational Research, **200**, pp. 385–394.

[102] Parikh PJ & Meller RD, 2008, *Selecting between batch and zone order picking strategies in a distribution center*, Transportation Research Part E, **44**, pp. 696–719.

[103] Park BC, Park JY & Foley RD, 2003, *Carousel system performance*, Journal of Applied Probability, **40 (3)**, pp. 602–612.

[104] Paul RJ & Thomas RC, 1977, *An integrated distribution, warehousing and inventory control system for imported books*, Operational Research Quarterly, **28 (3)**, pp. 629–640.

[105] PEP Stores, 2007, *PEP plans for growth*, [Online], [Cited June 24th, 2011], Available from `http://pepstores.com/who-we-are/latest-news/pep-plans-for-growth.html`

[106] Petersen CG, 2000, *An evaluation of order picking policies for mail order companies*, Production and Operations Management, **9 (4)**, pp. 319–335.

[107] Petersen CG & Aase G, 2004, *A comparison of picking, storage, and routing policies in manual order picking*, International Journal of Production Economics, **92**, pp. 11–19.

[108] Petersen CG & Schmenner RW, 1999, *An evaluation of routing and volume-based storage policies in an order picking operation*, Decision Sciences, **30 (2)**, pp. 481–501.

[109] Petersen CG, Siu C & Heiser DR, 2005, *Improving order picking performance utilizing slotting and golden zone storage*, International Journal of Operations & Production Management, **25 (10)**, pp. 997–1012.

[110] Psion, 2009, *Psion Teklogix PEP stores case study*, [Online], [Cited June 24th, 2011], Available from `http://www.psion.com`

[111] Renaud J & Ruiz A, 2008, *Improving product location and order picking activities in a distribution centre*, Journal of the Operational Research Society, **59**, pp. 1603–1613.

[112] Rim S-C & Park I-S, 2008, *Order picking plan to maximize the order fill rate*, Computers & Industrial Engineering, **55**, pp. 557–566.

[113] Roodbergen KJ and De Koster MBM, 2001, *Routing methods for warehouses with multiple cross aisles*, International Journal of Production Research, **39 (9)**, pp. 1865–1883.

[114] Roodbergen KJ & De Koster MBM, 2001, *Routing order pickers in a warehouse with a middle aisle*, European Journal Of Operational Research, **133**, pp. 32–43.

[115] Rouwenhorst B, Reuter B, Stockrahm V, Van Houtum GJ, Mantel RJ & Zijm WHM, 2000, *Warehouse design and control: Framework and literature review*, European Journal Of Operational Research, **122**, pp. 515–533.

[116] Ruben RA & Jacobs FR, 1999, *Batch construction heuristics and storage assignment strategies for walk/ride, and pick systems*, Management Science, **45 (4)** , pp. 575–596.

[117] Rubrico JIU, Higashi T, Tamura H & Ota J, 2011, *Online rescheduling of multiple picking agents for warehouse management*, Robotics and Computer Integrated Manufacturing, **27**, pp. 62–71.

[118] Samuels J, 2010, *Space allocation on a permanent picking line with flowracks*, HonsBComm project, University of Stellenbosch, Stellenbosch.

[119] Scott M, 2009, *Optimisation of a single-aisle picking line*, HonsBSc project, University of Stellenbosch, Stellenbosch.

[120] Sharp G & Krueger KW, 2000, *A simulation software tool for order picking in a person-aboard storage/retrieval system*, [Online], [Cited June 24$^{\text{th}}$, 2011], Available from `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.19.6793&rep=rep1&type=pdf`

[121] The South African Advertising Research Foundation, 2004, *Living Standards Measure*, [Online], [Cited June 24$^{\text{th}}$, 2011], Available from `http://www.saarf.co.za/LSM/lsms.asp`

[122] Sullivan D, 2006, *Searches per day*, [Online], [Cited June 24$^{\text{th}}$, 2011], Available from `http://searchenginewatch.com/article/2067276/Searches-Per-Day`

[123] Tang LC & Chew E-P, 1997, *Order picking systems: Batching and storage assignment strategies*, Computers & Industrial Engineering, **33 (3-4)**, pp. 817–820.

[124] Theys C, Bräysy O, Dullaert W and Raa B, 2010, *Using a TSP heuristic for routing order pickers in warehouses*, European Journal of Operational Research, **200**, pp. 755–763.

[125] Van Den Berg JP, 1999, *A literature survey on planning and control of warehousing systems*, IIE Transactions, **31**, pp. 751–762.

[126] Van Den Berg JP, 1996, *Multiple order pick sequencing in a carousel system: A solvable case of the rural postman problem*, Journal of the Operational Research Society, **47 (12)**, pp. 1504–1515.

[127] Van Den Berg JP, Sharp GP, Gademann AJRM & Pochet Y, 1998, *Forward-reserve allocation in a warehouse with unit-load replenishments*, European Journal of Operational Research, **111**, pp. 98–113.

[128] Van Den Berg JP & Zijm WHM, 1999, *Models for warehouse management: Classification and examples*, International Journal of Production Economics, **59**, pp. 519–528.

[129] Van Oudheusden DL, Tzen Y-JJ& Ko H-T, 1988, *Improving storage and order picking in a person-on-board AS/R system: A case study*, Engineering Costs and Production Economics, **13**, pp. 273–283.

[130] Vickson RG & Fujimoto A, 1996, *Optimal storage locations in a carousel storage and retrieval system*, Location Science, **4 (4)**, pp. 237–245.

[131] Vickson RG & Lu X, 1998, *Optimal product and server locations in one-dimensional storage racks*, European Journal Of Operational Research, **105**, pp. 18–28.

[132] Vitasek K, 2010, *Supply chain management terms and glossary*, [Online], [Cited June 24^{th}, 2011], Available from `http://cscmp.org/digital/glossary/glossary.asp`

[133] Wen U-P & Chang D-T, 1988, *Picking rules for a carousel conveyor in an automated warehouse*, Omega International Journal of Management Science, **16 (2)**, pp. 145–151.

[134] Wright J, 2010, *Towards smarter supply chains*, OECD Observer, **279**, pp. 44–45.

[135] XJ Technologies, AnyLogic 6.5.1, [Online], [Cited November 1^{st}, 2011], Available from `http://www.xjtek.com/`.

[136] Yu M & De Koster MBM, 2009, *The impact of order batching and picking area zoning on order picking system performance*, European Journal of Operational Research, **198**, pp. 480–490.

[137] Zoller K & Robrade A, 1988, *Efficient heuristics for dynamic lot sizing*, International Journal of Production Research, **26 (2)**, pp. 249–265.

APPENDIX A

Tables for algorithms

In Chapter 6 the various algorithms are presented along with the input files for each algorithm. The following tables are additions to those discussed in Chapter 6 and indicate the number of SKUs that are assigned to individual picking lines and the number of SKUs to duplicate for the various algorithms presented.

The formulas to calculate the variables required as part of the input files is presented in Chapter 6 and only the tables containing the variables if 3 of the original picking lines are selected. The tables containing the variables when 2, 4 and 5 of the original picking lines are selected are presented here. Tables A.1–A.3 contains the variables for Algorithm 1. The variables for Algorithm 2 is contained in Tables A.4–A.6. For Algorithms 7–11 the variables for the input files are equal and are contained in Tables A.7–A.9. For Algorithms 12–15 2 SKUs are assigned to a picking line simultaneously, thus the variables are slightly different from that of Algorithms 7–11 and are contained in Tables A.10–A.12.

Original picking line number	New picking lines			
	Picking line number	# SKUs	Duplications	Removals
1	1	38	18	18
2	2	38	18	18
–	3	36	20	0
1	1	28	28	28
2	2	28	28	28
–	3	28	28	0
–	4	28	28	0

Table A.1: *Number of SKUs to duplicate and remove per picking line when creating additional lines from 2 original picking lines for Algorithm 1, if all the selected picking lines contain 56 SKUs.*

Original picking line	New picking lines			
number	Picking line number	# SKUs	Duplications	Removals
1	1	45	11	11
2	2	45	11	11
3	3	45	11	11
4	4	45	11	11
–	5	44	12	0
1	1	38	18	18
2	2	38	18	18
3	3	38	18	18
4	4	38	18	18
–	5	36	20	0
–	6	36	20	0
1	1	32	24	24
2	2	32	24	24
3	3	32	24	24
4	4	32	24	24
–	5	32	24	0
–	6	32	24	0
–	7	32	24	0
1	1	28	28	28
2	2	28	28	28
3	3	28	28	28
4	4	28	28	28
–	5	28	28	0
–	6	28	28	0
–	7	28	28	0
–	8	28	28	0

Table A.2: *Number of SKUs to duplicate and remove per picking line when creating additional lines from 4 original picking lines for Algorithm 1, if all the selected picking lines contain 56 SKUs.*

Original picking line number	New picking lines			
	Picking line number	# SKUs	Duplications	Removals
1	1	47	9	9
2	2	47	9	9
3	3	47	9	9
4	4	47	9	9
5	5	47	9	9
–	6	45	11	0
1	1	40	16	16
2	2	40	16	16
3	3	40	16	16
4	4	40	16	16
5	5	40	16	16
–	6	40	16	0
–	7	40	16	0
1	1	35	21	21
2	2	35	21	21
3	3	35	21	21
4	4	35	21	21
5	5	35	21	21
–	6	35	21	0
–	7	35	21	0
–	8	35	21	0
1	1	32	24	24
2	2	32	24	24
3	3	32	24	24
4	4	32	24	24
5	5	32	24	24
–	6	30	26	0
–	7	30	26	0
–	8	30	26	0
–	9	30	26	0
1	1	31	25	25
2	2	31	25	25
3	3	31	25	25
4	4	31	25	25
5	5	31	25	25
–	6	25	25	0
–	7	25	25	0
–	8	25	25	0
–	9	25	25	0
–	10	25	25	0

Table A.3: *Number of SKUs to duplicate and remove per picking line when creating additional lines from 5 original picking lines for Algorithm 1, if all the selected picking lines contain 56 SKUs.*

Original picking line number	New picking lines		
	Picking line number	Duplications	Removals
1	1	28	28
2	2	28	28
–	3	0	0
1	1	56	56
2	2	56	56
–	3	0	0
–	4	0	0

Table A.4: *Number of SKUs to duplicate and remove per picking line when constructing new picking lines when selecting 2 of the original picking lines for Algorithm 2, if all of the selected original picking lines contained 56 SKUs.*

Original picking line number	New picking lines		
	Picking line number	Duplications	Removals
1	1	14	14
2	2	14	14
3	3	14	14
4	4	14	14
–	5	0	0
1	1	28	28
2	2	28	28
3	3	28	28
4	4	28	28
–	5	0	0
–	6	0	0
1	1	42	42
2	2	42	42
3	3	42	42
4	4	42	42
–	5	0	0
–	6	0	0
–	7	0	0
1	1	56	56
2	2	56	56
3	3	56	56
4	4	56	56
–	5	0	0
–	6	0	0
–	7	0	0
–	8	0	0

Table A.5: *Number of SKUs to duplicate and remove per picking line when constructing new picking lines when selecting 4 of the original picking lines for Algorithm 2, if all of the selected original picking lines contained 56 SKUs.*

Original picking line number	New picking lines		
	Picking line number	Duplications	Removals
1	1	11	11
2	2	11	11
3	3	11	11
4	4	11	11
5	5	11	11
–	6	1	0
1	1	22	22
2	2	22	22
3	3	22	22
4	4	22	22
5	5	22	22
–	6	1	0
–	7	1	0
1	1	33	33
2	2	33	33
3	3	33	33
4	4	33	33
5	5	33	33
–	6	1	0
–	7	1	0
–	8	1	0
1	1	44	44
2	2	44	44
3	3	44	44
4	4	44	44
5	5	44	44
–	6	1	0
–	7	1	0
–	8	1	0
–	9	1	0
1	1	55	55
2	2	55	55
3	3	55	55
4	4	55	55
5	5	55	55
–	6	1	0
–	7	1	0
–	8	1	0
–	9	1	0
–	10	1	0

Table A.6: *Number of SKUs to duplicate and remove per picking line when constructing new picking lines when selecting 5 of the original picking lines for Algorithm 2, if all of the selected original picking lines contained 56 SKUs.*

Nr picking lines (L)	Picking line (k)	# SKUs (c_k)	Duplications (d_k)
3	1	38	18
	2	37	19
	3	37	19
4	1	28	28
	2	28	28
	3	28	28
	4	28	28

Table A.7: *Number of SKUs to be assigned to each new picking line and the number of duplications required to fill all the locations on each line, when selecting 2 of the original picking lines and applying Algorithms 7–11, if all the original picking lines contain 56 SKUs.*

Nr picking lines (L)	Picking line (k)	# SKUs (c_k)	Duplications (d_k)
5	1	45	11
	2	45	11
	3	45	11
	4	45	11
	5	44	12
6	1	38	18
	2	38	18
	3	37	19
	4	37	19
	5	37	19
	6	37	19
7	1	32	24
	2	32	24
	3	32	24
	4	32	24
	5	32	24
	6	32	24
	7	32	24
	8	32	24
8	1	28	28
	2	28	28
	3	28	28
	4	28	28
	5	28	28
	6	28	28
	7	28	28
	8	28	28

Table A.8: *Number of SKUs to be assigned to each new picking line and the number of duplications required to fill all the locations on each line, when selecting 4 of the original picking lines and applying Algorithms 7–11, if all the original picking lines contain 56 SKUs.*

Nr picking lines (L)	Picking line (k)	# SKUs (c_k)	Duplications (d_k)
6	1	47	9
	2	47	9
	3	47	9
	4	47	9
	5	46	10
	6	46	10
7	1	40	16
	2	40	16
	3	40	16
	4	40	16
	5	40	16
	6	40	16
	7	40	16
8	1	35	21
	2	35	21
	3	35	21
	4	35	21
	5	35	21
	6	35	21
	7	35	21
	8	35	21
9	1	32	24
	2	31	25
	3	31	25
	4	31	25
	5	31	25
	6	31	25
	7	31	25
	8	31	25
	9	31	25
10	1	28	28
	2	28	28
	3	28	28
	4	28	28
	5	28	28
	6	28	28
	7	28	28
	8	28	28
	9	28	28
	10	28	28

Table A.9: *Number of SKUs to be assigned to each new picking line and the number of duplications required to fill all the locations on each line, when selecting 5 of the original picking lines and applying Algorithms 7–11, if all the original picking lines contain 56 SKUs.*

Nr picking lines (L)	Picking line (k)	# SKUs (c_k)	Duplications (d_k)
3	1	38	18
	2	38	18
	3	36	20
4	1	28	28
	2	28	28
	3	28	28
	4	28	28

Table A.10: *Number of SKUs to be assigned to each new picking line and the number of duplications required to fill all the locations on each line, when selecting 2 of the original picking lines and applying Algorithms 12–15, if all the original picking lines contain 56 SKUs.*

Nr picking lines (L)	Picking line (k)	# SKUs (c_k)	Duplications (d_k)
5	1	46	10
	2	46	10
	3	44	12
	4	44	12
	5	44	12
6	1	38	18
	2	38	18
	3	38	18
	4	38	18
	5	36	20
	6	36	20
7	1	32	24
	2	32	24
	3	32	24
	4	32	24
	5	32	24
	6	32	24
	7	32	24
	8	32	24
8	1	28	28
	2	28	28
	3	28	28
	4	28	28
	5	28	28
	6	28	28
	7	28	28
	8	28	28

Table A.11: *Number of SKUs to be assigned to each new picking line and the number of duplications required to fill all the locations on each line, when selecting 4 of the original picking lines and applying Algorithms 12–15, if all the original picking lines contain 56 SKUs.*

Nr picking lines (L)	Picking line (k)	# SKUs (c_k)	Duplications (d_k)
6	1	48	8
	2	48	8
	3	46	10
	4	46	10
	5	46	10
	6	46	10
7	1	40	16
	2	40	16
	3	40	16
	4	40	16
	5	40	16
	6	40	16
	7	40	16
8	1	36	20
	2	36	20
	3	36	20
	4	36	20
	5	34	22
	6	34	22
	7	34	22
	8	34	22
9	1	32	24
	2	32	24
	3	32	24
	4	32	24
	5	32	24
	6	30	26
	7	30	26
	8	30	26
	9	30	26
10	1	28	28
	2	28	28
	3	28	28
	4	28	28
	5	28	28
	6	28	28
	7	28	28
	8	28	28
	9	28	28
	10	28	28

Table A.12: *Number of SKUs to be assigned to each new picking line and the number of duplications required to fill all the locations on each line, when selecting 5 of the original picking lines and applying Algorithms 12–15, if all the original picking lines contain 56 SKUs.*

APPENDIX B

Additional pseudo codes

In Chapter 6 the various algorithms are presented. The pseudo codes for Algorithms 14 and 15 are similar to that of Algorithms 12 and 13 respectively and are therefore given here. Algorithm 14 (PC/C5/D) only differ from Algorithm 12 in order in which the picking lines are filled, with Algorithm 14 implementing M1 (cyclical assignment) and Algorithm 12 implementing M2 (SLSS assignment). The same is true for Algorithm 13 and 15, with Algorithm 15 implementing M1 (cyclical assignment) and Algorithm 13 implementing M2 (SLSS assignment).

Algorithm 14: PC/C5/D

Input : The variables L, C and c_k, sets $\mathcal{S}, \mathcal{L}$ and $\mathcal{P}$ and similarity matrix $\mathbf{X}$.
Output : Sets $\mathcal{S}^*, \mathcal{L}^*$ and $\mathcal{P}^*$.

$i \leftarrow 1$;
$j \leftarrow i+1$;
for $k = 1$ *to* L **do**
  for $i = 1$ *to* n **do**
    for $j = i+1$ *to* n **do**
      if x_{ij} *is the maximum* **then**
        $\mathcal{S}_k \leftarrow \mathcal{S}_k \cup s_i \cup s_j$;
        $\mathcal{L}_k \leftarrow \mathcal{L}_k \cup l_i \cup l_j$;
        $\mathcal{P}_k \leftarrow \mathcal{P}_k \cup p_i \cup p_j$;
        Remove SKUs i and j from similarity matrix $\mathbf{X}$;
      end
    end
  end
end
for $k = 1$ *to* L **do**
  if $C > c_k$ **then**
    Sort SKUs in set $\mathcal{S}_k$ from highest pick frequency per location to lowest for picking line k;
    $j \leftarrow 1$;
    while $C > |\mathcal{L}_k|_\Sigma$ **do**
      $l_{kj} \leftarrow 2$;
      $j = j+1$;
    end
  end
end
$\mathcal{S}^* \leftarrow \mathcal{S}_1 \cup \mathcal{S}_2 \cup \ldots \cup \mathcal{S}_L$;
$\mathcal{L}^* \leftarrow \mathcal{L}_1 \cup \mathcal{L}_2 \cup \ldots \cup \mathcal{L}_L$;
$\mathcal{P}^* \leftarrow \mathcal{P}_1 \cup \mathcal{P}_2 \cup \ldots \cup \mathcal{P}_L$;

Algorithm 15: PC/C6/D

Input : The variables L, C and c_k, sets $\mathcal{S}, \mathcal{L}$ and $\mathcal{P}$ and dissimilarity matrix $\mathbf{Y}'$.
Output : Sets $\mathcal{S}^*, \mathcal{L}^*$ and $\mathcal{P}^*$.

$i \leftarrow 1$;
$j \leftarrow i + 1$;
$k \leftarrow 1$;
for $k = 1$ *to* L **do**
  for $i = 1$ *to* n **do**
    for $j = i + 1$ *to* n **do**
      if y'_{ij} *is the minimum* **then**
        $\mathcal{S}_k \leftarrow \mathcal{S}_k \cup s_i \cup s_j$;
        $\mathcal{L}_k \leftarrow \mathcal{L}_k \cup l_i \cup l_j$;
        $\mathcal{P}_k \leftarrow \mathcal{P}_k \cup p_i \cup p_j$;
        Remove SKUs i and j from dissimilarity matrix $\mathbf{Y}'$;
      end
    end
  end
end
for $k = 1$ *to* L **do**
  if $C > c_k$ **then**
    Sort SKUs in set $\mathcal{S}_k$ from highest pick frequency per location to lowest for picking line k;
    $j \leftarrow 1$;
    while $C > |\mathcal{L}_k|_\Sigma$ **do**
      $l_{kj} \leftarrow 2$;
      $j = j + 1$;
    end
  end
end
$\mathcal{S}^* \leftarrow \mathcal{S}_1 \cup \mathcal{S}_2 \cup \ldots \cup \mathcal{S}_L$;
$\mathcal{L}^* \leftarrow \mathcal{L}_1 \cup \mathcal{L}_2 \cup \ldots \cup \mathcal{L}_L$;
$\mathcal{P}^* \leftarrow \mathcal{P}_1 \cup \mathcal{P}_2 \cup \ldots \cup \mathcal{P}_L$;

APPENDIX C

Results

In Chapter 7 the results achieved by implementing the various algorithms are discussed. In this appendix more detailed results are presented in addition to the figures and tables presented in Chapter 7. This gives more detail concerning the saving in the number of cycles achieved and the percentage of SKUs that are duplicated in order to achieved these results.

Figures C.1–C.15 gives the number of cycles that is required to complete the order picking for all the groups for the various algorithms respectively. Only Figures C.2, C.5 and C.7 is given in Chapter 7 as Algorithms 2, 5 and 7 respresent Groups A2, A1 and A3 respectively. Figure C.16 gives the biggest percentage saving in the number of cycles for each Groups A-5 to V-2 when implementing each of the algorithms in addition to Figure 7.1 in Chapter 7 which only gives the average percentage saving for the groups consisting of 2, 3, 4 and 5 of the original picking lines. Figure C.17 gives the percentage SKUs that are duplicated to achieve the biggest percentage saving in the number of cycles for each Groups A-5 to V-2 when implementing each of the algorithms in addition to Figure 7.3 in Chapter 7 which only gives the average percentage SKUs duplicated for the groups consisting of 2, 3, 4 and 5 of the original picking lines.

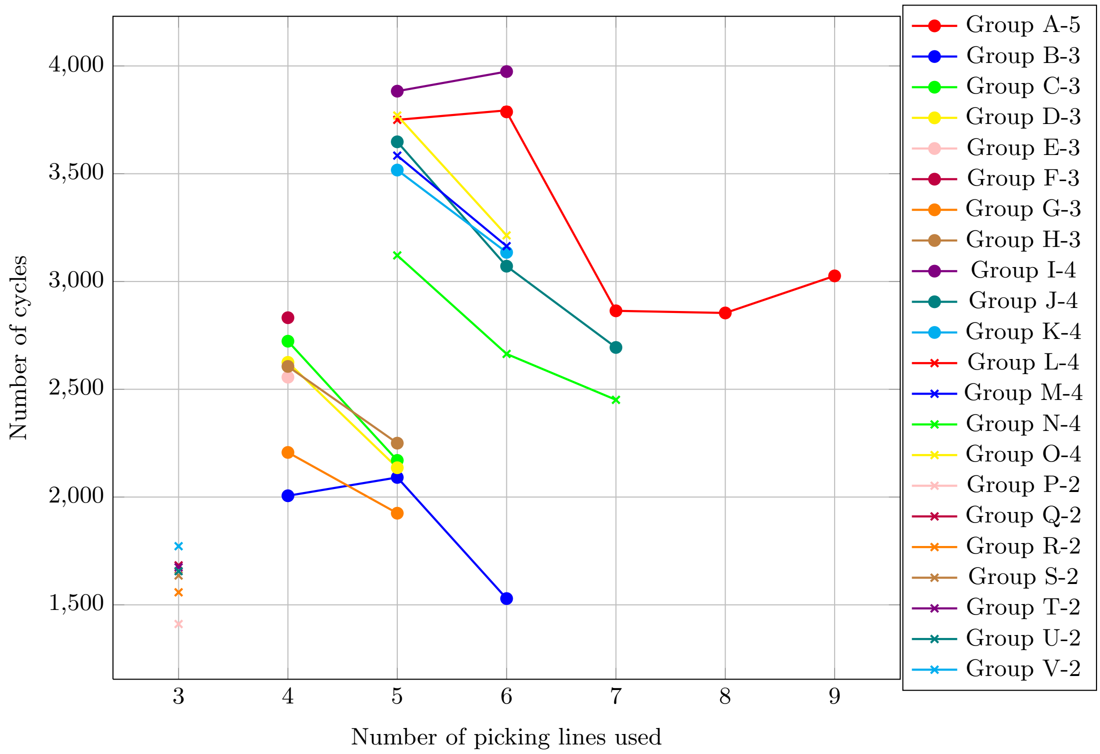


Figure C.1: *Comparison of the number of cycles to complete the branch orders for all the groups for Algorithm 1 (PS/D/M1-ND) with an increase in the number of picking lines.*

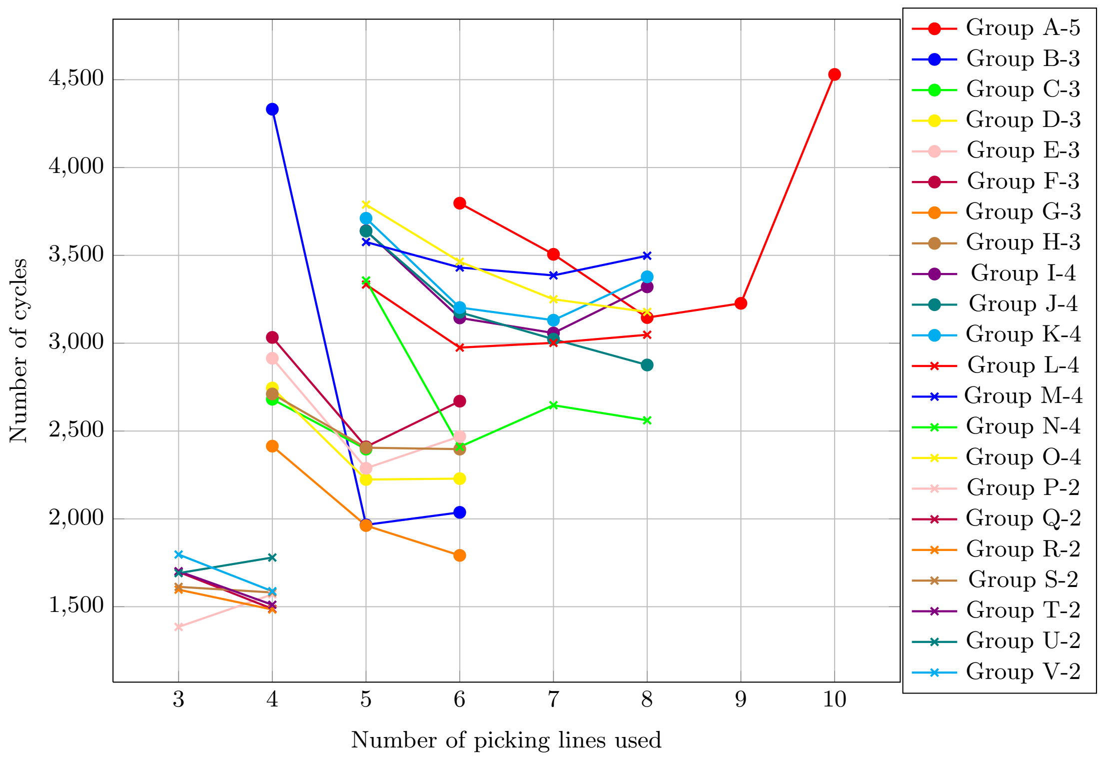


Figure C.2: *Comparison of the number of cycles to complete the branch orders for all the groups for Algorithm 2 (PS/D/M1-S) with an increase in the number of picking lines.*

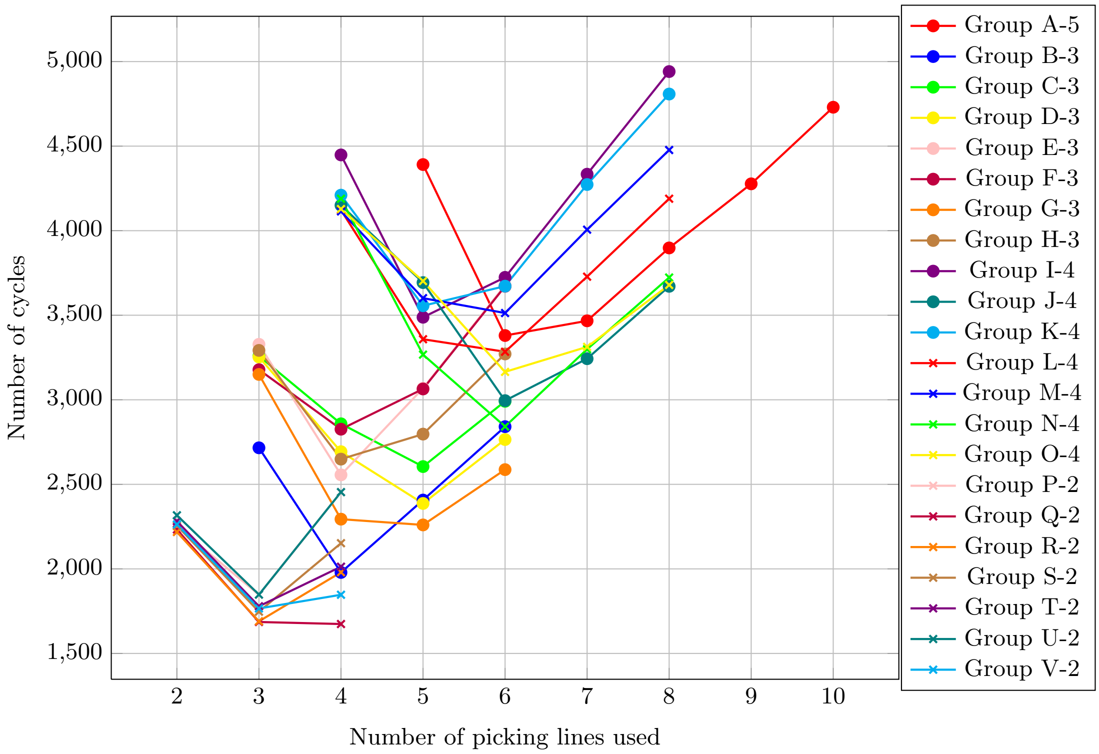


Figure C.3: *Comparison of the number of cycles to complete the branch orders for all the groups for Algorithm 3 (PC/D/M1-ND) with an increase in the number of picking lines.*

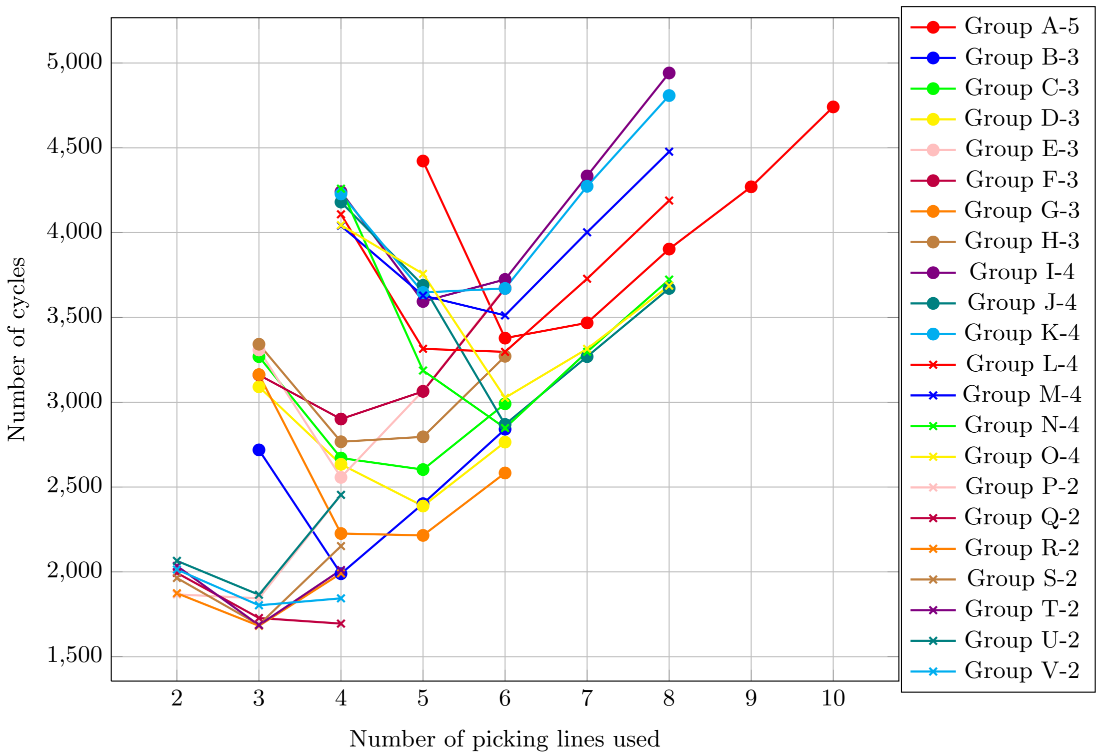


Figure C.4: *Comparison of the number of cycles to complete the branch orders for all the groups for Algorithm 4 (PC/D/M1-S) with an increase in the number of picking lines.*

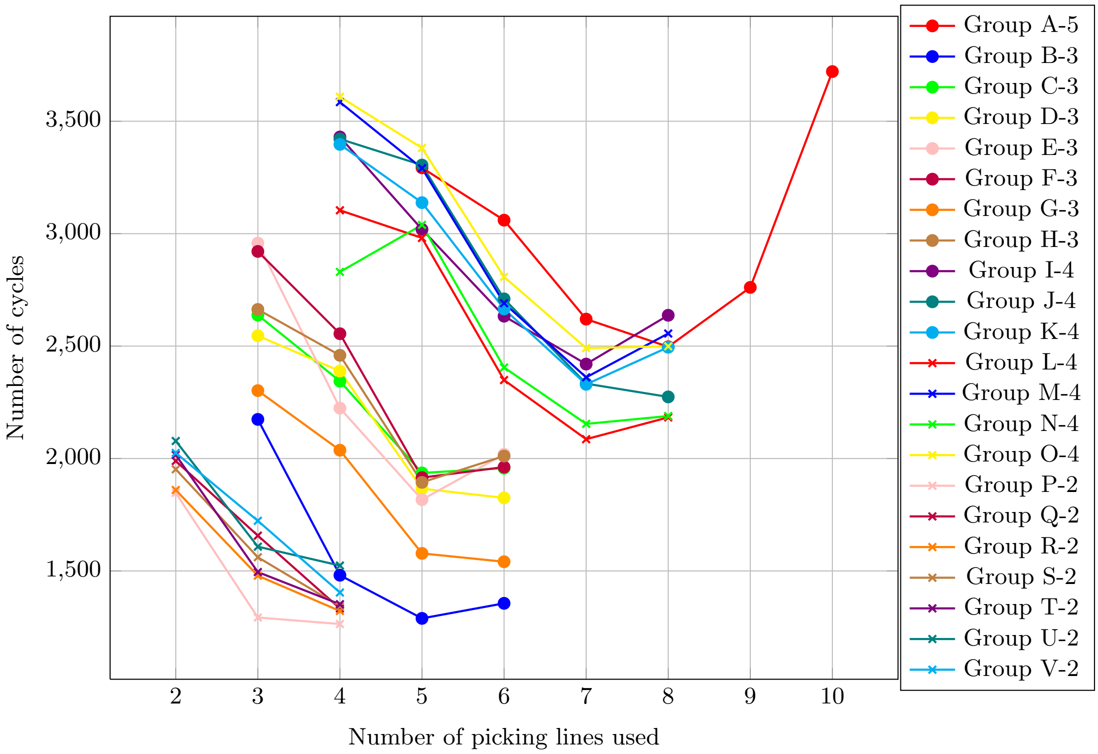


Figure C.5: *Comparison of the number of cycles to complete the branch orders for all the groups for Algorithm 5 (PC/D/M2-S) with an increase in the number of picking lines.*

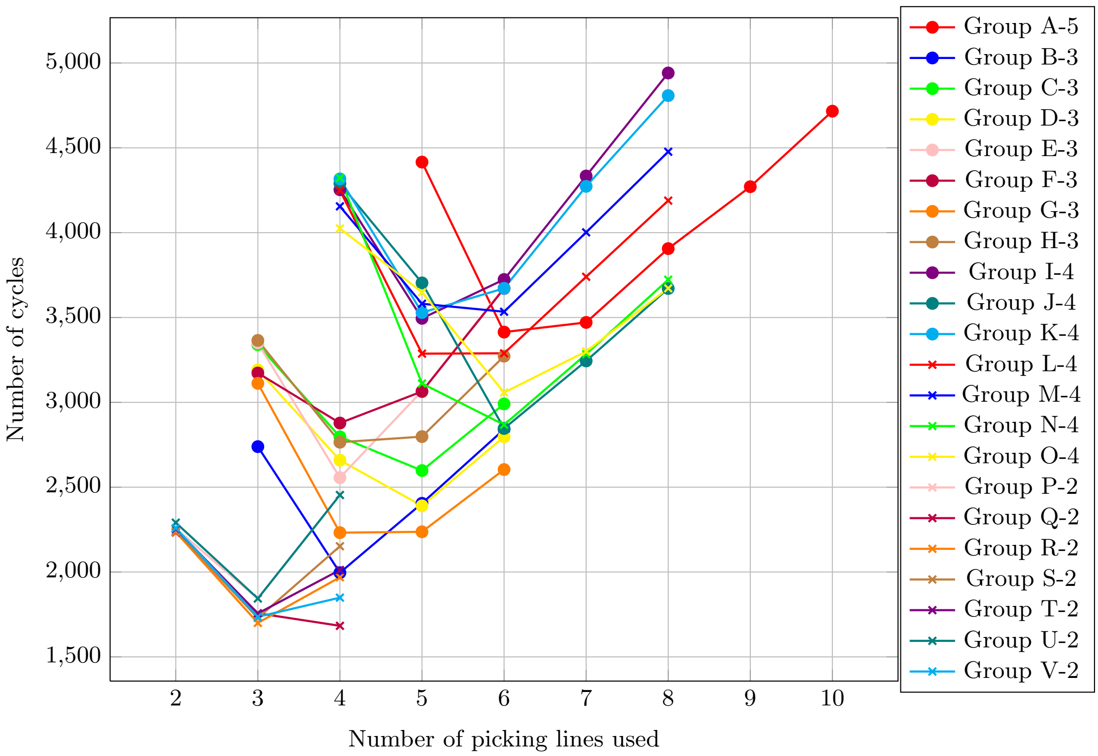


Figure C.6: *Comparison of the number of cycles to complete the branch orders for all the groups for Algorithm 6 (PC/D/M3-S) with an increase in the number of picking lines.*

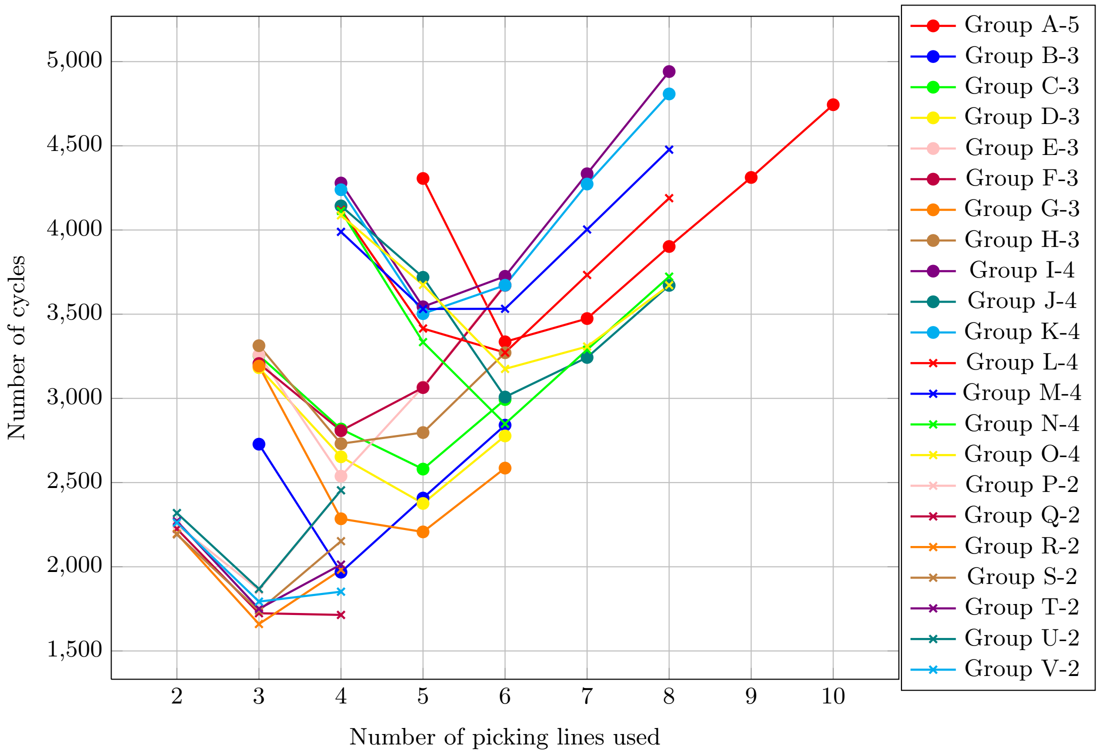


Figure C.7: *Comparison of the number of cycles to complete the branch orders for all the groups for Algorithm 7 (PC/M1/D) with an increase in the number of picking lines.*

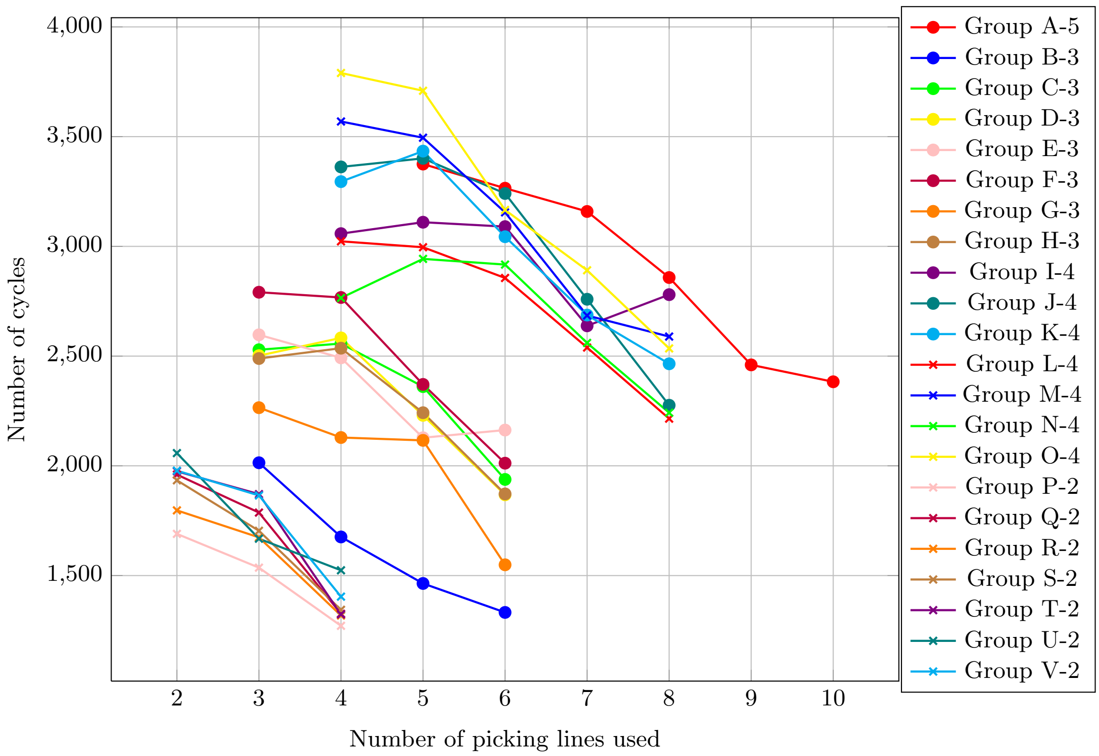


Figure C.8: *Comparison of the number of cycles to complete the branch orders for all the groups for Algorithm 8 (PC/M2/D) with an increase in the number of picking lines.*

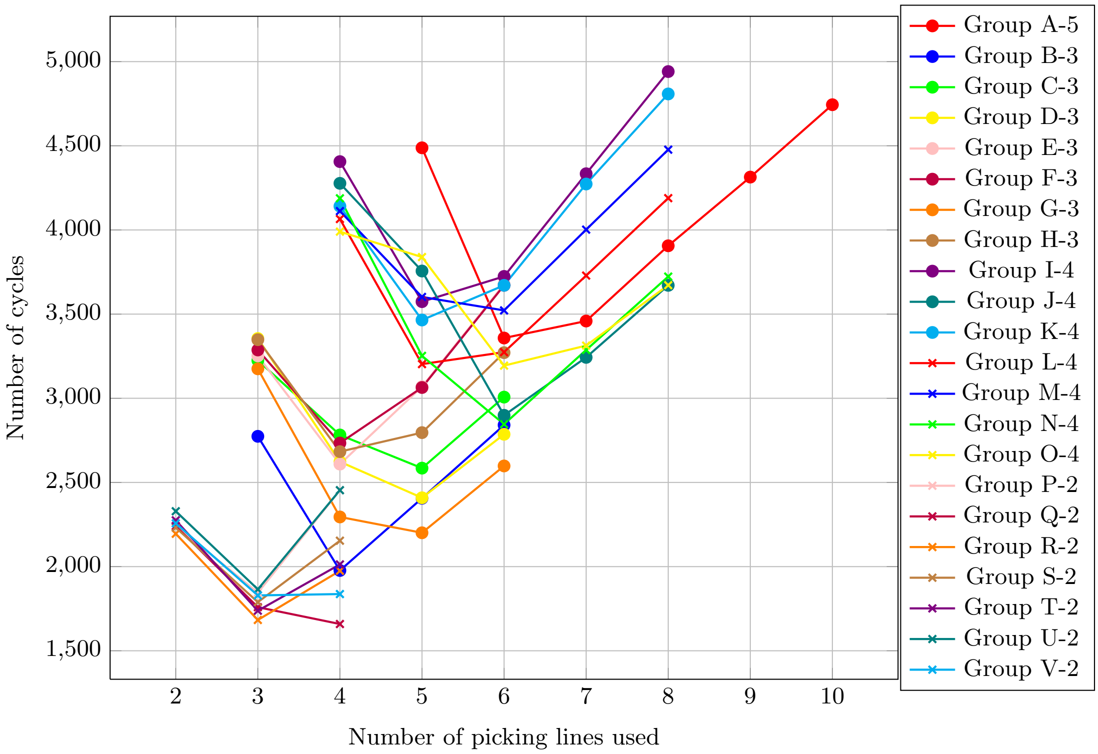


Figure C.9: *Comparison of the number of cycles to complete the branch orders for all the groups for Algorithm 9 (PC/M3/D) with an increase in the number of picking lines.*

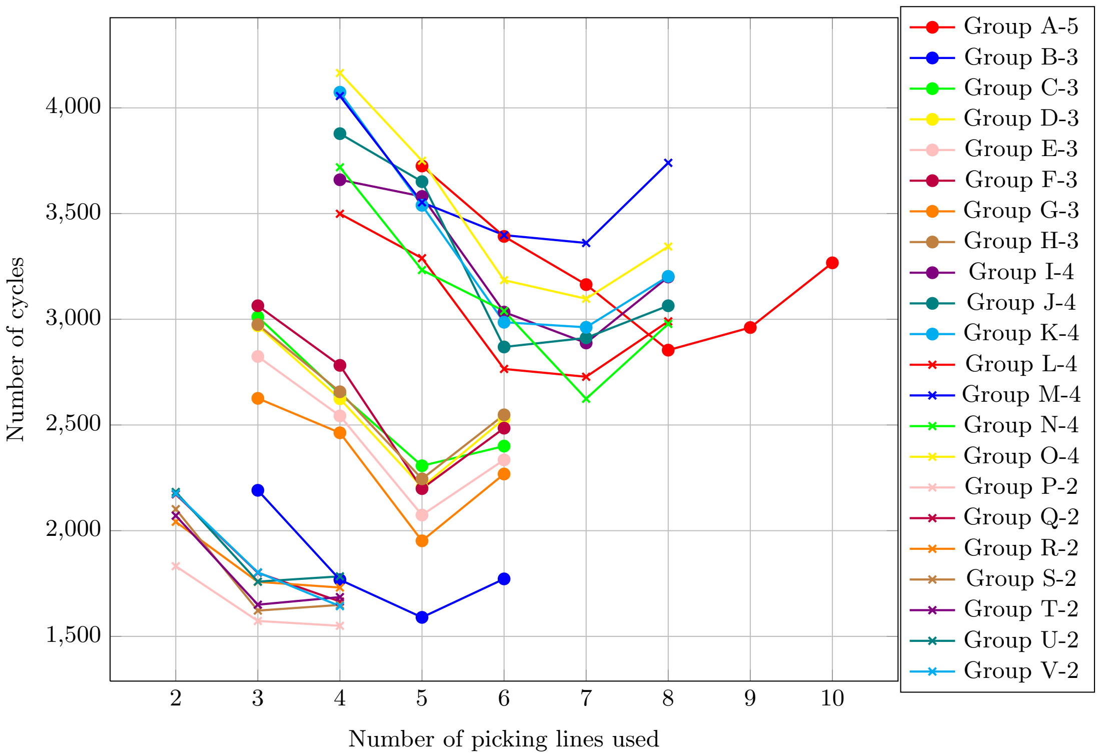


Figure C.10: *Comparison of the number of cycles to complete the branch orders for all the groups for Algorithm 10 (PC/C1/D) with an increase in the number of picking lines.*

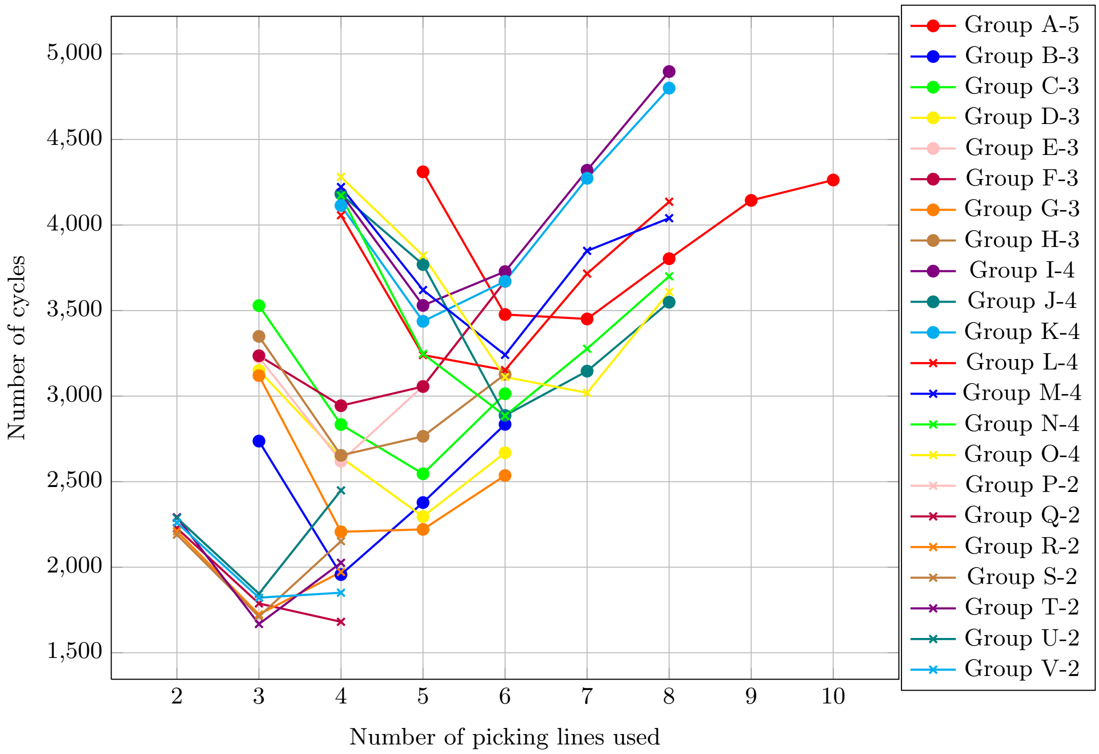


Figure C.11: *Comparison of the number of cycles to complete the branch orders for all the groups for Algorithm 11 (PC/C2/D) with an increase in the number of picking lines.*

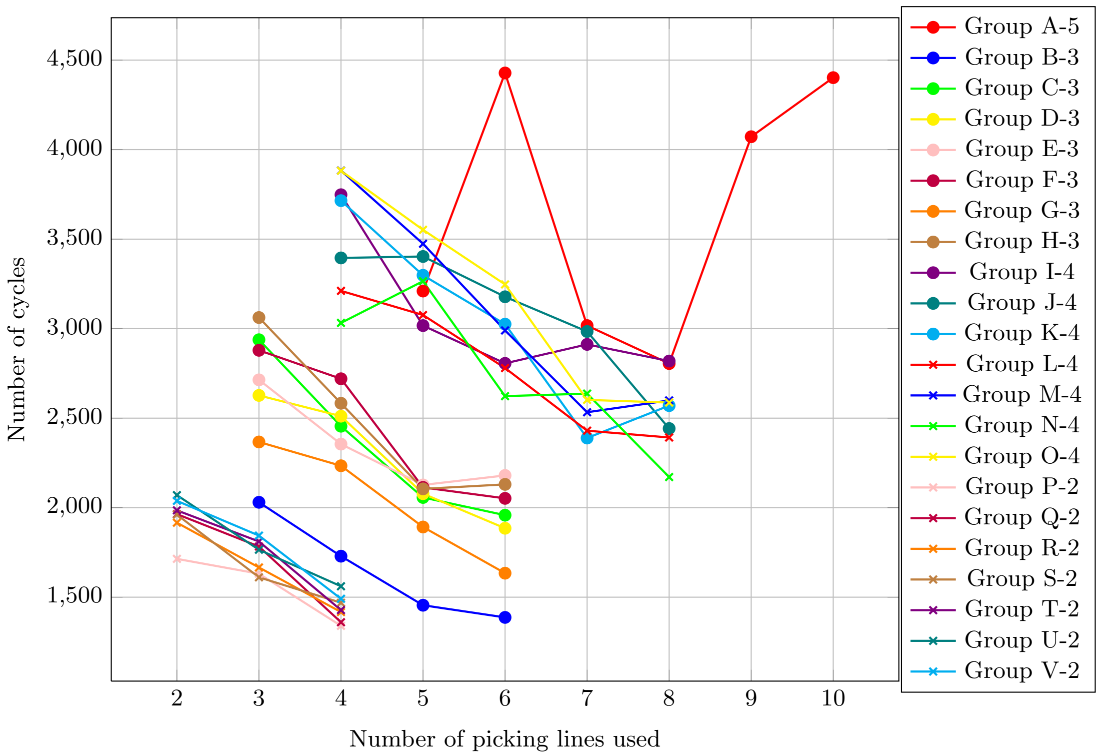


Figure C.12: *Comparison of the number of cycles to complete the branch orders for all the groups for Algorithm 12 (PC/C3/D) with an increase in the number of picking lines.*

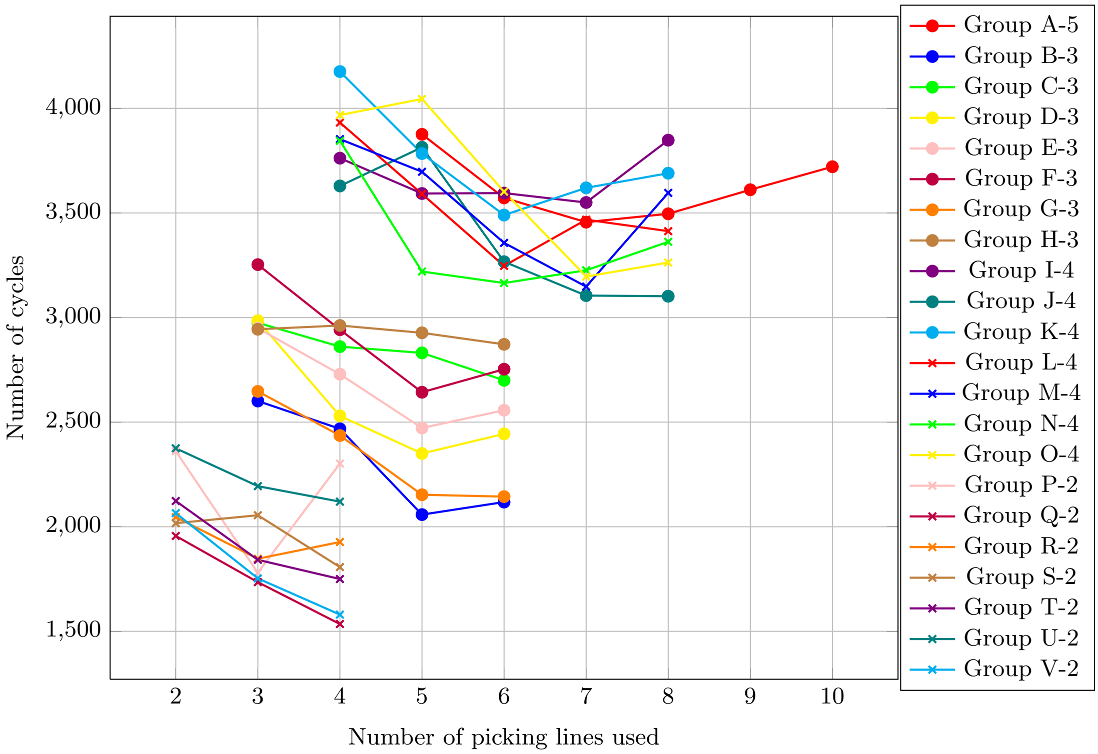


Figure C.13: *Comparison of the number of cycles to complete the branch orders for all the groups for Algorithm 13 (PC/C4/D) with an increase in the number of picking lines.*

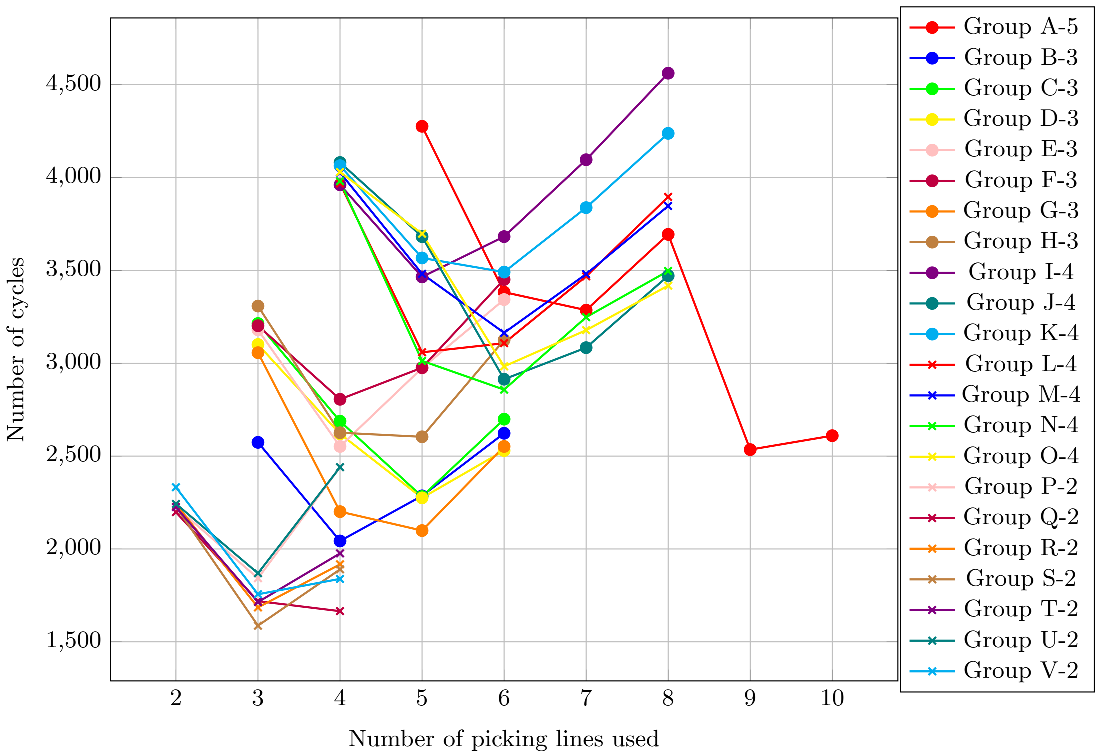


Figure C.14: *Comparison of the number of cycles to complete the branch orders for all the groups for Algorithm 14 (PC/C5/D) with an increase in the number of picking lines.*

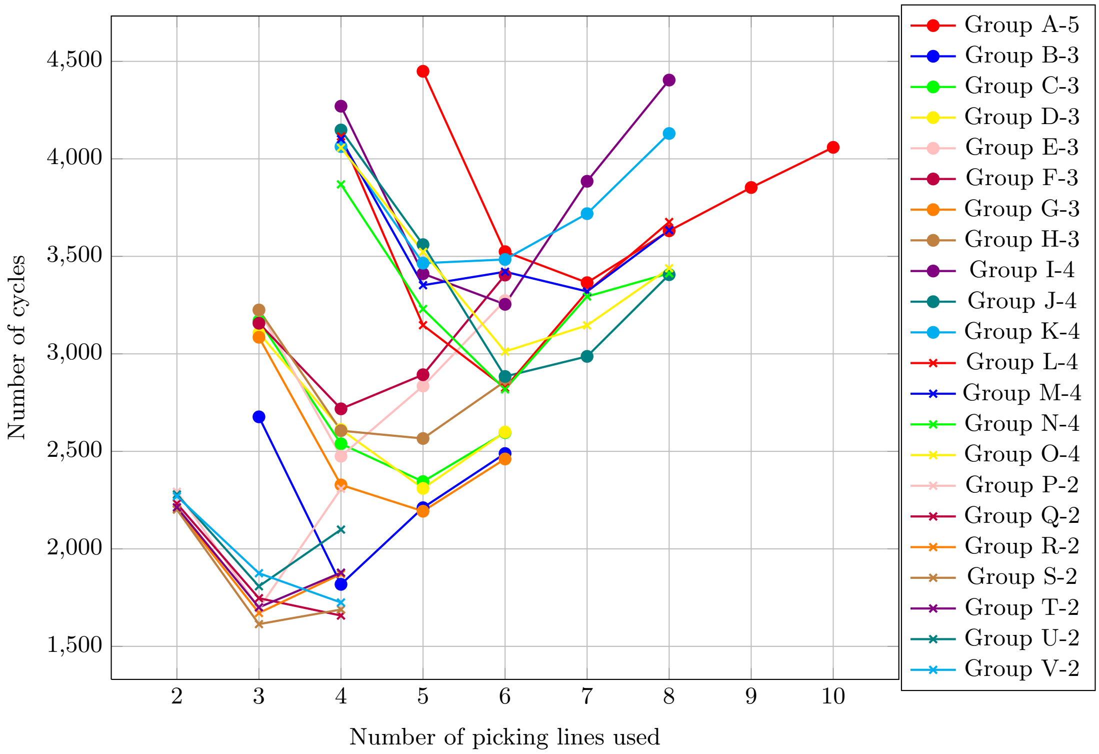


Figure C.15: *Comparison of the number of cycles to complete the branch orders for all the groups for Algorithm 15 (PC/C6/D) with an increase in the number of picking lines.*

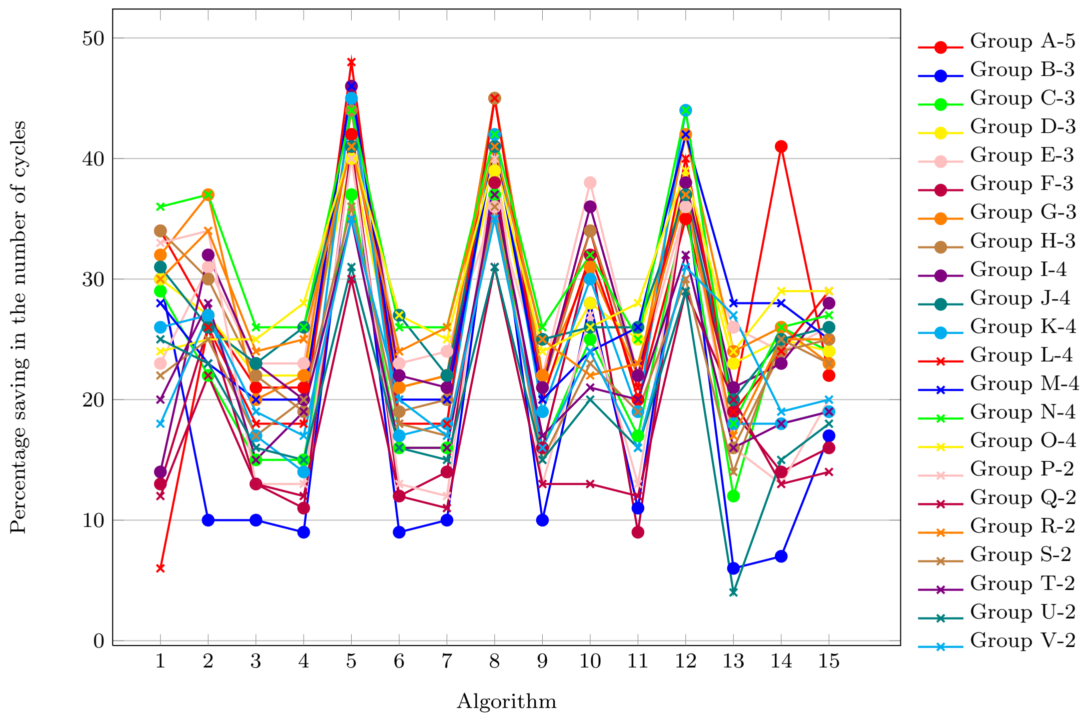


Figure C.16: *Percentage saving in the minimum number of cycles required for each group when implementing the various algorithms presented in Chapter 6 as in Table 7.2.*

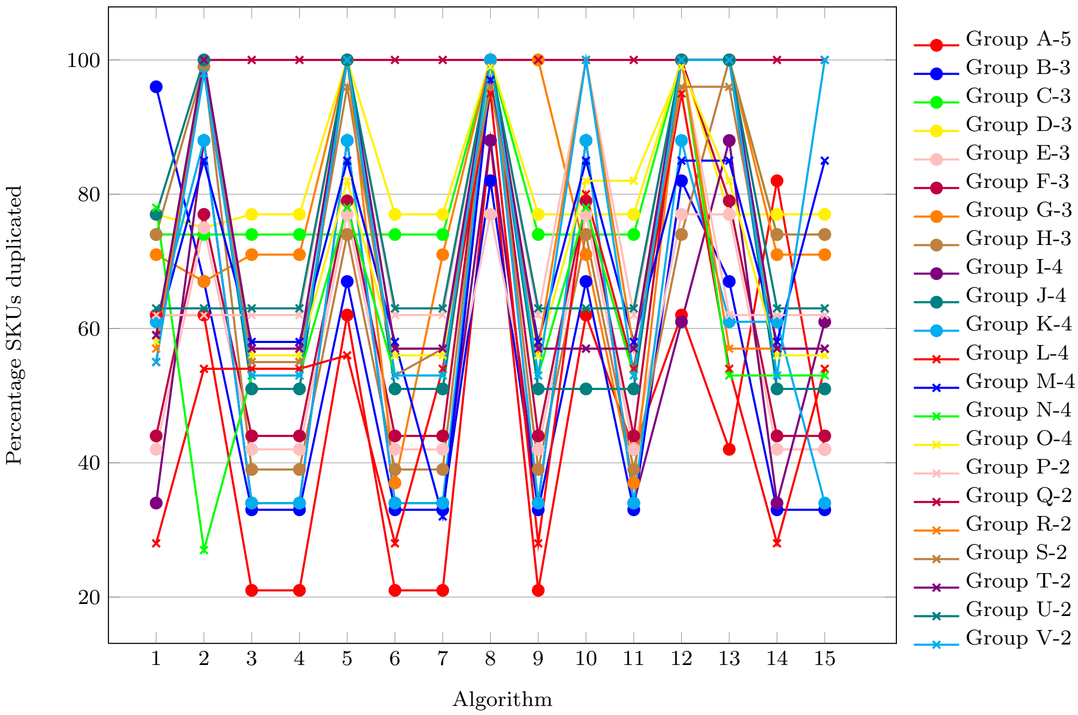


Figure C.17: *Percentage SKUs duplicated for each group at which the minimum total number of cycles is achieved when implementing the various algorithms presented in Chapter 6 as in Table 7.2.*